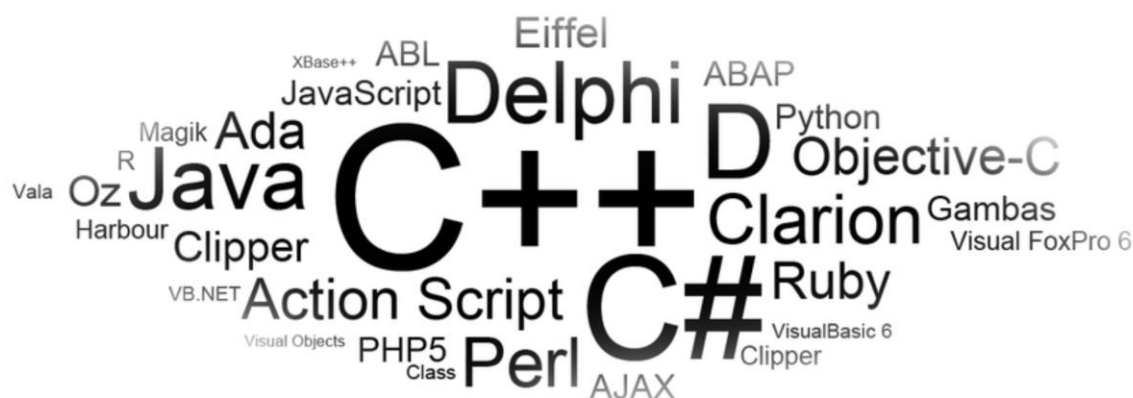


# ДАСТУРЛАШ АСОСЛАРИ



**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ  
ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ**

**МИРЗО УЛУГ'БЕК НОМИДАГИ ЎЗБЕКИСТОН МИЛЛИЙ  
УНИВЕРСИТЕТИ**



**“Дастурлаш асослари”  
фани бўйича  
Ў Қ У В – У С Л У Б И Й М А Ж М У А**

**Тошкент – 2016**

Мазкур ўқув-услугий мажмуа Олий ва ўрта махсус таълим вазирлигининг 2016 йил 6 апрелидаги 137-сонли буйруғи билан тасдиқланган ўқув режа ва дастур асосида тайёрланди.

Тузувчилар: ЎзМУ доценти А.Икрамов  
ЎзМУ ўқитувчиси Қ.Махаров

Такризчилар: ТАТУ профессори А.Акбаров  
ЎзМУ профессори А.Ахмедов

***Ўқув -услугий мажмуа ЎзМУ кенгашининг 2016 йил \_\_\_\_\_ даги \_\_\_\_ -  
сонли қарори билан тасдиққа тавсия қилинган.***

## МУНДАРИЖА

<b>СЎЗ БОШИ.....</b>	<b>6</b>
<b>I. СИЛЛАБУС .....</b>	<b>8</b>
<b>II. 1. ИШЧИ ЎҚУВ РЕЖА.....</b>	<b>14</b>
<b>II. 2. НАМУНАВИЙ ВА ИШЧИ ЎҚУВ ДАСТУР.....</b>	<b>16</b>
<b>III. МОДУЛНИ ЎҚИТИШДА ФОЙДАЛАНИЛАДИГАН ИНТРЕФАОЛ ТАЪЛИМ МЕТОДЛАРИ .....</b>	<b>37</b>
<b>I СЕМЕСТР .....</b>	<b>40</b>
<b>IV. МАЪРУЗА МАТЕРИАЛЛАРИ .....</b>	<b>41</b>
<b>Маъруза 1. С++ тили синтаксиси ва унинг лексик асоси.....</b>	<b>41</b>
<b>Маъруза 2. С++ тили дастурининг тузилиши ва шакли.....</b>	<b>52</b>
<b>Маъруза 3. Берилганлар турлари. С++ тилининг таянч турлари.....</b>	<b>64</b>
<b>Маъруза 4. Ўзгарувчилар ва ифодалар. ....</b>	<b>75</b>
<b>Маъруза 5. Амаллар: инкремент, декремент, sizeof, мантикий,     разрядли, такқослаш. Амалларнинг устунликлари ва бажарилиш     йўналишлари.....</b>	<b>91</b>
<b>Маъруза 6. Ўқиш-ёзиш оқимлари (cin, cout).....</b>	<b>105</b>
<b>Маъруза 7-8. Операторлар. Шарт операторлари. ....</b>	<b>122</b>
<b>Маъруза 9-10. Такрорлаш операторлари. Бошқарувни узатиш     операторлари.....</b>	<b>152</b>
<b>Маъруза 11-12. Статик массивлар.....</b>	<b>186</b>
<b>Маъруза 13-14. Функциялар эълон қилиш ва аниқлаш. main()     функцияси. Локал ва глобал ўзгарувчилар. ....</b>	<b>216</b>
<b>Маъруза 15. Рекурсив функциялар. ....</b>	<b>249</b>
<b>Маъруза 16. Фойдаланувчи томонидан аниқланган берилганлар     турлари.....</b>	<b>265</b>
<b>Маъруза 17. Стандарт кутубхона функциялари.....</b>	<b>282</b>

Маъруза 18. Кўрсаткичлар ва адрес олувчи ўзгарувчилар. ....	297
Маъруза 19. Динамик массивлар.....	314
Маъруза 20-21. Сатрлар. Сатр устида амаллар. Сатр функциялари... ..	329
<b>V. АМАЛИЙ МАШҒУЛОТ МАТЕРИАЛЛАРИ.....</b>	<b>357</b>
Амалий машғулот 1. C++ тили синтаксиси ва унинг лексик асоси .....	357
Амалий машғулот 2. C++ тили дастурининг тузилиши ва шакли .....	365
Амалий машғулот 3. Берилганлар турлари. C++ тилининг таянч турлари.....	371
Амалий машғулот 4. Ўзгарувчилар ва ифодалар .....	378
Амалий машғулот 5-6. Амаллар: инкремент, декремент, sizeof, мантиқий, разрядли, таққослаш.....	384
Амалий машғулот 7. Ўқиш-ёзиш оқимлари (cin, cout) .....	392
Амалий машғулот 8-9. Операторлар. Шарт операторлари.....	399
Амалий машғулот 10-11. Такрорлаш операторлари. Бошқарувни узатиш операторлари .....	409
Амалий машғулот 12-13. Статик массивлар .....	420
Амалий машғулот 14-15. Функциялар эълон қилиш ва аниқлаш. main() функцияси. Локал ва глобал ўзгарувчилар .....	429
Амалий машғулот 16. Рекурсив функциялар .....	440
Амалий машғулот 17. Фойдаланувчи томонидан аниқланган берилганлар турлари.....	449
Амалий машғулот 18. Стандарт кутубхона функциялари .....	457
Амалий машғулот 19. Кўрсаткичлар ва адрес олувчи ўзгарувчилар..	468
Амалий машғулот 20-21. Динамик массивлар.....	476
Амалий машғулот 22-23. Сатрлар. Сатр устида амаллар. Сатр функциялари .....	486
<b>II СЕМЕСТР .....</b>	<b>496</b>
<b>IV. МАЪРУЗА МАТЕРИАЛЛАРИ .....</b>	<b>497</b>

Маъруза 22. Функция ва массивлар. ....	497
Маъруза 23-24. Тузилмалар. Бирлашмалар. ....	515
Маъруза 25-26. Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ўқиш ва ёзиш функциялари. ....	538
Маъруза 27-28. Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари. ....	556
Маъруза 29-30. Динамик тузилмалар. ....	578
<b>V. АМАЛИЙ МАШҒУЛОТ МАТЕРИАЛЛАРИ.....</b>	<b>600</b>
Амалий машғулот 24-25. Функция ва массивлар .....	600
Амалий машғулот 26-27. Тузилмалар. Бирлашмалар .....	611
Амалий машғулот 28-29. Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ўқиш ва ёзиш функциялари .....	623
Амалий машғулот 30-31. Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари .....	634
Амалий машғулот 32-33. Динамик тузилмалар.....	644

## СЎЗ БОШИ

Мазкур ўқув услубий мажмуа “Дастурлаш асослари” фанидан “5130100-Математика” таълим йўналиши учун мўлжалланган бўлиб, Математика факультетининг “Алгоритмлар ва дастурлаш технологиялари” кафедраси профессор-ўқитувчилари томонидан ишлаб чиқилган. “Дастурлаш асослари” фани ўқув услубий мажмуасини яратишда етакчи хорижий ОТМлари ўқув дастурларига асосий адабиётлар рўйхатига киритилган D.S. Malik C++ Programming: From Problem Analysis to Program Design (Course Technology, 2011), Ivor Horton Beginning Visual C++ 2005 (Wiley Publishing, 2005) адабиётлардан фойдаланилди.

“Дастурлаш асослари” фани “5130100-Математика” таълим йўналиши ўқув режасига асосан 1- ва 2-семестрларда мос равишда 90 ва 36 аудитория соатларда ўқитилади. 1-семестрда C++ тили ҳақида умумий тушунчалар, дастур тузилиши, дастурни ёзиш шакли, берилганларнинг турлари, ифодаларнинг ёзилиши, амаллар, шарт, такрорлаш операторлари, бир ва кўп ўлчамли массивлар билан ишлаш асослари, статик ва динамик массивлар билан ишлаш, функциялар, функция кўринишлари, сатрлар тузилиши, сатрлар билан ишлаш функциялари тўғрисида умумий маълумотлар келтирилган, намунавий мисоллар асосида тушунтирилган. 2-семестрда функция ва массивлардан биргаликда фойдаланиш, тузилмалар, бирлашмалар, матн ва бинар файллар билан ишлаш асослари, файлдан фойдаланиб дастур тузишда ишлатиладиган асосий ва керакли функцияларнинг кўриниши ва ишлатилиши, берилганларнинг динамик тузилмалари, чизикли рўйхат, стек ва навбат ҳақида маълумотлар берилган, улар билан ишлаш намунавий мисоллар асосида тушунтириб ўтилган.

“Дастурлаш асослари” фанини ўзлаштириш жараёнида бакалавр ахборот, уни сақлаш усуллари, қайта ишлаш ва узатиш, ҳисоблаш тизимларининг математик ва дастурий таъминоти, уларни фан соҳаларида, ишлаб чиқариш ва таълимда қўллаш хусусиятлари, компьютерни дастурий

таъминоти, дастур турлари ва хусусиятлари, структурали, дастурни оптималлаштириш ва умумлаштириш, дастурлашда модулли тамойилларини қўллаш, компьютер технологиялари ютуқларини замонавий ҳисоблаш тизимларининг математик ва дастурий таъминотида қўллаш, юқори даражадаги дастурлаш тилларини, дастурий таъминотни, дастурлаш технологияларини, татбиқий ва ҳисоблаш математикаси масалаларини ечиш алгоритмларини, модулли таҳлил ва модулли дастурлаш асосларини, самарали дастур ва дастурлар комплексини яратиш усулларини билиш ва улардан фойдалана олиш, татбиқий масалаларни ечиш алгоритминини тузиш, математик (компьютер) моделини куриш ва унинг дастурий таъминотини яратиш, кўникмаларига эга бўлади.

Ушбу ўқув услубий қўлланма бешта қисмдан иборат бўлиб, улар силлабус, ишчи ўқув режа, намунавий ва ишчи ўқув дастур, модулни ўқитишда фойдаланиладиган интерфаол таълим методлари, маъруза материаллари (маъруза матни, адабиётлар рўйхати, мустақил таълим мавзулари, глоссарий, кейслар банки, назорат саволлари ва тест саволлари) ва амалий машғулотлар материаллари (амалий топшириқлар, намуна, адабиётлар рўйхати, тарқатма материаллар, кейслар банки, тест саволлари)дан ташкил топган. Маъруза ва амалий машғулотлар материаллари семестрларга ажратилган ҳолда берилган.



# I. СИЛЛАБУС

## «Дастурлаш асослари» фанининг силлабуси

(2016/2017 ўқув йили)

<b>Кафедра номи:</b>	<b>Программалаш ва тармоқ технологиялари</b>	
<b>Ўқитувчи ҳақида маълумот:</b>	Доцент Икромов Аҳмад Маорипович Ўқитувчи Махаров Қодирбек Толқунович	ikramov_am@nuuz.uz baron9090@mail.ru
<b>Семестр ва ўқув курсининг давомийлиги</b>	1,2 - Семестр ва жами соат	
<b>Ўқув соатлари ҳажми:</b>	<b>жами:</b>	212
	шунингдек:	
	маъруза	60
	семинар	-
	Амалий (лаборатория)	66
	мустақил таълим	86
<b>Йўналиш номи ва шифри</b>	5130100	Математика

**Курснинг предмети ва мазмуни:** Фан назарий ва амалий қисмлардан иборат. Назарий қисм информатика ва ҳисоблаш техникаси, алгоритмлар, C/C++ дастурлаш тили, C++Builder объектга йўналтирилган дастурлаш муҳитларида ишлаш бўйича кўрсатмалар бўлимларидан ташкил топган.

Дастурда компьютерда дастурлашга киришнинг назарий асоси бўлган алгоритмларга алоҳида эътибор қаратилган. Бу ерда алгоритмларни тавсифлаш ва кейинчалик компьютерда амалга ошириш учун зарур бўлган бир қатор математик тушунчалар - такрорлаш, ёрдамчи алгоритм, рекурсия, хотира, массив, индекс, функция, параметр ва ҳ.к. киритилиб, турли хил синф масалаларининг алгоритмлари тузилади.

Дастурлаш тили - тузилган алгоритмни компьютер амалга ошириши учун воситадир. Бу ўринда турли мураккабликдаги синтаксис ва семантикага эга бўлган тиллардан фойдаланиш мумкин.

**Курсни ўқитишнинг мақсади ва вазифалари:** Фанни ўқитишдан мақсад – “Математика” йўналишининг бакалавр босқичи талабаларига дастурлаш асосларини етарли даражада ўқитиш, шу билимларга таянган ҳолда компьютер ёрдамида моделлаштиришга келадиган тадбиқий масалаларнинг дастур таъминотини амалга оширишга ўргатиш ва ихтисослик фанларини ўзлаштиришда таянч билимларга эга бўлиш.

Фаннинг вазифалари – масала ечишнинг алгоритмик асосларини ўрганиш, компьютер ишлашининг тамоили, дастурлаш тилларини синфлаш, компьютерда берилганлар ва буйруқларни тасвирланиши, C++ тилида дастурлаш, объектга йўналтирилган дастурлаш технологиялари, визуал дастурлаш муҳитида ишлаш бу фаннинг асосий вазифалари ҳисобланади.

### Курснинг таркиби ва мазмуни

№	Мавзулар	Маъруза	Амалий (семинар)	Мустақил иш
1-семестр				
1.	C++ тили синтаксиси ва унинг лексик асоси.	2	2	2
2.	C++ тили дастурининг тузилиши ва шакли	2	2	2
3.	Берилганлар турлари. C++ тилининг таянч турлари.	2	2	2
4.	Ўзгарувчилар ва ифодалар.	2	2	2
5.	Амаллар: инкремент, декремент, sizeof, мантқиқий, разрядли, таққослаш. Амалларнинг устунликлари ва бажарилиш йўналишлари.	2	4	4
6.	Ўқиш-ёзиш оқимлари (cin, cout).	2	2	4
7.	Операторлар. Шарт операторлари.	4	4	4
8.	Такрорлаш операторлари. Бошқарувни узатиш операторлари.	4	4	6
9.	Статик массивлар.	4	4	6
10.	Функциялар эълон қилиш ва аниқлаш. main() функцияси. Локал ва глобал ўзгарувчилар.	4	4	6

11.	Рекурсив функциялар.	2	2	4
12.	Фойдаланувчи томонидан аниқланган берилганлар турлари.	2	2	2
13.	Стандарт кутубхона функциялари.	2	2	4
14.	Кўрсаткичлар ва адрес олувчи ўзгарувчилар.	2	2	2
15.	Динамик массивлар.	2	4	6
16.	Сатрлар. Сатр устида амаллар. Сатр функциялари.	4	4	4
2 семестр				
17.	Функция ва массивлар.	2	4	4
18.	Тузилмалар. Бирлашмалар.	4	4	6
19.	Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ўқиш ва ёзиш функциялари.	4	4	6
20.	Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари.	4	4	4
21.	Динамик тузилмалар.	4	4	6
<b>Жами:</b>		<b>60</b>	<b>66</b>	<b>86</b>

**1-маъруза. C++ тили синтаксиси ва унинг лексик асоси.**

Дастурлаш тиллари. C++ тилидаги содда программа. C++ тилида изоҳлар. C++ тили алфавити ва лексемалари.

**2-маъруза. C++ тили дастурининг тузилиши ва шакли.**

Тил синтаксиси. Тил семантикаси. Хужжатлашган идентификаторлар. Дастурни форматлаш

**3-маъруза. Берилганлар турлари. C++ тилининг таянч турлари.**

Берилганларнинг оддий турлари. Белги тури. Мантикий тур. Ҳақиқий сон тури

**4-маъруза. Ўзгарувчилар ва ифодалар.**

Таянч арифметик амаллар. Турни бошқа турга келтириш. Константа ўзгармаслар. Ифодаларда ўзгарувчилардан фойдаланиш

**5-маъруза. Амаллар: инкремент, декремент, sizeof, мантикий, разрядли, таққослаш.**

Амалларнинг устунликлари ва бажарилиш йўналишлари.

Инкремент ва декремент амаллари. sizeof амали. Разрядли мантикий амаллар. Чапга ва ўнгга суриш амаллари.

Амалларнинг устунликлари ва бажарилиш йўналишлари

**6-маъруза. Ўқиш-ёзиш оқимлари (cin, cout).**

Оқимлар. Ўқиш оқими (cin). Ёзиш оқими (cout)

**7-8-маъруза. Операторлар. Шарт операторлари.**

Таққослаш амаллари. Мантикий операторлар. Шарт операторлари. if оператори. if...else оператори. ?: шарт амали. switch оператори

**9-10-маъруза. Такрорлаш операторлари. Бошқарувни узатиш операторлари.**

Такрорланувчи жараёнлар. for такрорлаш оператори. while такрорлаш оператори. do-while такрорлаш оператори. break оператори. continue оператори. goto оператори ва нишонлар

**11-12-маъруза. Статик массивлар.**

Массивлар ҳақида тушунча. Массив элементига мурожаат қилиш. Кўп ўлчамли массивлар

Бир ўлчамли массивларни қайта ишлаш. Бир ўлчамли массивларни инициализация қилиш.

Кўп ўлчамли массивларни инициализациялаш

**13-14-маъруза. Функциялар эълон қилиш ва аниқлаш. main() функцияси. Локал ва глобал ўзгарувчилар.**

Функциялардан фойдаланиш. Олдиндан аниқланган функциялар. Фойдаланувчи томонидан аниқланадиган

функциялар. Келишув бўйича аргументлар. Қўриниш соҳаси. Локал ва глобал ўзгарувчилар. :: амали.

Жойлаштириладиган (inline) функциялар. Қайта юкланувчи функциялар.

**15-маъруза. Рекурсив функциялар.**

Рекурсия тушунчаси. «Ханой минораси» масаласи. Фибоначчи сонларини топиш масаласи.

**16-маъруза. Фойдаланувчи томонидан аниқланган берилганлар турлари.**

Санаб ўтилувчи тур. Санаб ўтилувчи тур устида амаллар бажариш. Функциялар ва enum турлар. typedef билан ишлаш.

**17-маъруза. Стандарт кутубхона функциялари.**

сctype кутубхона файли. Математик функциялар кутубхонаси. Белгилар билан ишлаш функциялари.

Турларни ўзгартириш функциялари.

**18-маъруза. Кўрсаткичлар ва адрес олувчи ўзгарувчилар.**

Кўрсаткичлар. Кўрсаткичга бошланғич қиймат бериш. Кўрсаткич устида амаллар. Адресни олиш амали.

Кўрсаткичлар ва адрес олувчи ўзгарувчилар функция параметри сифатида.

**19-маъруза. Динамик массивлар.**

new оператори. delete оператори. Динамик массивлар билан ишлаш. new оператори ёрдамида динамик массивлар билан

**20-21-маъруза. Сатрлар. Сатр устида амаллар. Сатр функциялари.**

ASCIIZ-сатрлар. ASCIIZ-сатрлар узунлигини аниқлаш функциялари. ASCIIZ-сатрларни нусхалаш. ASCIIZ-сатрларни улаш. ASCIIZ-сатрларда излаш функциялари. string туридаги сатрлар. Сатр қисмини бошқа сатрга нусхалаш функцияси. Сатр қисмини ўчириш ва алмаштириш функциялари. Сатр қисмини излаш ва солиштириш функциялари

**22-маъруза. Функция ва массивлар.**

Бир ўлчамли массив функция параметри сифатида. Кўп ўлчамли массив функция параметри сифатида. Ўзгарувчан параметрли функциялар

**23-24-маъруза. Тузилмалар. Бирлашмалар.**

Структуралар. Структура функция аргументи сифатида. Структуралар массиви. Структураларга кўрсаткич. Бирлашмалар ва улар устида амаллар

**25-26-маъруза. Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ўқиш ва ёзиш функциялари.**

Файл тушунчаси. Матн файллар. Бинар файллар. Файл оқимлари билан ишлаш. Файлга форматли ёзиш функциялари. Файлдан форматли ўқиш функциялари.

**27-28-маъруза. Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари.**

Файлга ёзиш функциялари. Файлдан ўқиш функциялари. Файл кўрсаткичини бошқариш функциялари.

**29-30-маъруза. Динамик тузилмалар.**

Берилганларнинг динамик тузилмалари. Рўйхатлар. Икки томонлама рўйхатлар. Рўйхатдан излаш. Рўйхатга элемент қўшиш.

**Мустақил таълим:**

Талаба мустақил таълимнинг асосий мақсади – ўқитувчининг раҳбарлиги ва назоратида муайян ўқув ишларини мустақил равишда бажариш учун билим ва кўникмаларини шакллантириш ва ривожлантириш.

Мустақил ишларни бажариш жараёнида талабалар қуйидаги ишларни бажаради:

- дарслик ва ўқув қўлланмалар асосида фан мавзулари бўйича назарий тайёргарлик кўриш, амалий ва лаборатория машғулотларига тайёрланиш;
- тарқатма материаллар бўйича маърузаларни чуқур ўзлаштириш;
- фан мазмунида кўрсатилмаган дастурлаш тиллари ва муҳитлари билан танишиш ва қиёсий таҳлил қилиш;
- масофавий таълим орқали дастурлаш билан турдош фанлар бўйича ўқув курсларида қатнашиш ва мос сертификатларга эга бўлиш тавсия қилинади.

Талаба мустақил ишини ташкил этишда қуйидаги шакллардан фойдаланади:

- берилган мавзулар бўйича ахборот (реферат) тайёрлаш;
- назарий билимларни амалиётда қўллаш;
- макет, модел ва намуналар яратиш;
- илмий мақола, анжуманга маъруза тайёрлаш ва ҳ.к.

**Тавсия этилаётган мустақил ишларнинг мавзулари**

1. C ва C++ тили синтаксислари.
2. C++ ва бошқа тилларда (Pascal, C#) модулли дастурлаш.
3. Стандарт кутубхона. Оқим синфлари.
4. Алгоритмлар кутубхонаси. Сонли ҳисоблаш функцияларининг кутубхонаси.
5. .NET технологияси, C# тилини хусусиятлари.
6. 8,16,32 разрядли процессорлар.
7. Турли муҳитларда яратилган дастур объектларини боғлаш.
8. C++ тили учун консол режимидаги дастурлаш муҳитлари.
9. Динамик структураларни қайта ишлаш алгоритмлари.
10. Саралаш ва излашнинг самарали алгоритмлари.
11. C++ тили стандартлари.
12. C++ тили асосида яратилган тиллар (C# ва Java тиллари).

*Изоҳ:* Мустақил таълим соатлари ҳажмларидан келиб чиққан ҳолда ишчи дастурда мазкур мавзулар ичидан мустақил таълим мавзулари шакллантирилади.

<b>Маслаҳатлар ва топшириқларни топшириш вақти</b>	Чоршанба	9:00	Математика факультети, А-409																															
<b>Билимларни баҳолаш усуллари, мезонлари, ва тартиби:</b>																																		
<b>Баҳолаш усуллари</b>	<b>Ўзлаштириш назорати (1-семестр)</b>																																	
	<table border="1"> <thead> <tr> <th>№</th> <th>Рейтинг назорат / шакли, максимал баллари</th> <th>1-ЖН</th> <th>2-ЖН</th> <th>3-ЖН (МТ)</th> <th>1-ОН</th> <th>2-ОН</th> <th>ЯН</th> <th>Баллар йиғиндиси</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>Максимал балл</td> <td>10</td> <td>15</td> <td>15</td> <td>15</td> <td>15</td> <td>30</td> <td rowspan="3" style="text-align: center; vertical-align: middle;">100</td> </tr> <tr> <td>2.</td> <td>Шакли:</td> <td>Оғзаки</td> <td>Оғзаки</td> <td>Оғзаки</td> <td>Ёзма</td> <td>Ёзма</td> <td>Ёзма</td> </tr> <tr> <td>3.</td> <td>Муддати (ўқув йили ҳафталарида)</td> <td>8</td> <td>13</td> <td>17</td> <td>9</td> <td>16</td> <td>19</td> </tr> </tbody> </table>	№	Рейтинг назорат / шакли, максимал баллари	1-ЖН	2-ЖН	3-ЖН (МТ)	1-ОН	2-ОН	ЯН	Баллар йиғиндиси	1.	Максимал балл	10	15	15	15	15	30	100	2.	Шакли:	Оғзаки	Оғзаки	Оғзаки	Ёзма	Ёзма	Ёзма	3.	Муддати (ўқув йили ҳафталарида)	8	13	17	9	16
№	Рейтинг назорат / шакли, максимал баллари	1-ЖН	2-ЖН	3-ЖН (МТ)	1-ОН	2-ОН	ЯН	Баллар йиғиндиси																										
1.	Максимал балл	10	15	15	15	15	30	100																										
2.	Шакли:	Оғзаки	Оғзаки	Оғзаки	Ёзма	Ёзма	Ёзма																											
3.	Муддати (ўқув йили ҳафталарида)	8	13	17	9	16	19																											
<b>Баҳолаш мезонлари</b>	<b>Ўзлаштириш назорати (2-семестр)</b>																																	
<table border="1"> <thead> <tr> <th>№</th> <th>Рейтинг назорат / шакли, максимал баллари</th> <th>1-ЖН</th> <th>2-ЖН (МТ)</th> <th>1-ОН</th> <th>ЯН</th> <th>Баллар йиғиндиси</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>Максимал балл</td> <td>20</td> <td>30</td> <td>20</td> <td>30</td> <td rowspan="3" style="text-align: center; vertical-align: middle;">100</td> </tr> <tr> <td>2.</td> <td>Шакли:</td> <td>Оғзаки</td> <td>Оғзаки</td> <td>Ёзма</td> <td>Ёзма</td> </tr> <tr> <td>3.</td> <td>Муддати (ўқув йили ҳафталарида)</td> <td>32</td> <td>40</td> <td>35</td> <td>41</td> </tr> </tbody> </table>	№	Рейтинг назорат / шакли, максимал баллари	1-ЖН	2-ЖН (МТ)	1-ОН	ЯН	Баллар йиғиндиси	1.	Максимал балл	20	30	20	30	100	2.	Шакли:	Оғзаки	Оғзаки	Ёзма	Ёзма	3.	Муддати (ўқув йили ҳафталарида)	32	40	35	41								
№	Рейтинг назорат / шакли, максимал баллари	1-ЖН	2-ЖН (МТ)	1-ОН	ЯН	Баллар йиғиндиси																												
1.	Максимал балл	20	30	20	30	100																												
2.	Шакли:	Оғзаки	Оғзаки	Ёзма	Ёзма																													
3.	Муддати (ўқув йили ҳафталарида)	32	40	35	41																													
<p><i>а) 86-100 балл учун қуйидагиларга жавоб бериши лозим:</i></p> <ul style="list-style-type: none"> <li>– турли санок системаларида арифметик амаллар бажара олса;</li> <li>– алгоритмлар ҳақида тўлиқ тушунчаларга эга бўлса;</li> <li>– ихтиёрий қийинчиликдаги масалани блок-схемаларини туза олса;</li> <li>– С++ тилида турли қийинчиликдаги масалаларни еча билса;</li> <li>– С++ тилида кўрсаткичлар ва мурожаатлар билан ишлай олса;</li> <li>– С++ тилида статик ва динамик массивлар, структуралар, функциялар ва рекурсив функциялар билан бемалол ишлай олса;</li> <li>– С++ тилида макрослар билан ишлай олса;</li> <li>– С++ тилида матнли ва бинар файллар билан бемалол ишлай олса;</li> <li>– мустақил топшириқларни ўз вақтида, тўлиқ ва аниқ бажарган бўлса.</li> </ul> <p><i>б) 71-85 балл учун қуйидагиларга жавоб бериши лозим:</i></p> <ul style="list-style-type: none"> <li>– асосий санок системаларида арифметик амаллар бажара олса;</li> <li>– алгоритмлар ҳақида тушунчаларга эга бўлса;</li> <li>– қийин бўлмаган масаланинг блок-схемасини туза олса;</li> <li>– С++ тилида қийин бўлмаган масалаларни еча билса;</li> <li>– С++ тилида кўрсаткичлар ва мурожаатлар билан ишлай олса;</li> <li>– С++ тилида массивлар, структуралар, функциялар билан ишлай олса;</li> <li>– С++ тилида макрослар билан китобдан фойдаланган ҳолда</li> </ul>																																		

	<p>ишлай олса;</p> <ul style="list-style-type: none"> <li>- С++ тилида матнли ва бинар файллар билан китобдан фойдаланган ҳолда ишлай олса;</li> <li>- мустақил топшириқларни тўлиқ ва аниқ бажарган бўлса.</li> </ul> <p><i>в) 55-70 балл учун қуйидагиларга жавоб бериши лозим:</i></p> <ul style="list-style-type: none"> <li>- иккилик санок системасида арифметик амаллар бажара олса;</li> <li>- алгоритмлар ҳақида тушунчаларга эга бўлса;</li> <li>- оддий масалани блок-схемалар кўринишидаги алгоритмини туза олса;</li> <li>- С++ тилида оддий масалаларни еча билса;</li> <li>- С++ тилида массивлар, функцияларга оид топшириқларни китобдан фойдаланган ҳолда ишлай олса;</li> <li>- С++ тилида структуралар, макрослар, матнли ва бинар файллар ҳақида тушунчаларга эга бўлса;</li> <li>- мустақил топшириқларни бажарган бўлса;</li> <li>- фан бўйича саволларга мужмал ва чалкаш жавоблар олинса</li> </ul> <p><i>г) қуйидаги ҳолларда 0-54 балл билан баҳоланиши мумкин:</i></p> <ul style="list-style-type: none"> <li>- санок системаларида арифметик амаллар бажара олмаса;</li> <li>- алгоритмлар ҳақида тушунчаларга эга бўлмаса;</li> <li>- С++ тилида массивлар, структуралар, функциялар билан ишлай олмаса;</li> <li>- С++ тилида макрослар, файллар ҳақида тушунчаларга эга бўлмаса;</li> <li>- мустақил топшириқларни бажара олмаса;</li> <li>- фаннинг мазмунини билмаса.</li> </ul>
<b>Ахборот ресурс база: ЎзМУ Ресурс маркази компьютер синфлари</b>	
<b>Асосий адабиётлар:</b>	<ol style="list-style-type: none"> <li>1. Bjarne Stroustrup. The C++ Programming Language (3th Edition). Addison-Wesley, 1997.</li> <li>2. D.S. Malik. C++ Programming: From Problem Analysis to Program Design. Fifth Edition. Course Technology, 2011.</li> <li>3. Мадрахимов Ш.Ф., Гайназаров С.М. С++ тилида дастурлаш асослари// Тошкент, ЎзМУ, 2009, 196 бет.</li> <li>4. Madraximov Sh.F., Ikramov A.M., Babajanov M.R. C++ tilida programmalash bo'yicha masalalar to'plami. O'quv qo'llanma // Toshkent, O'zbekiston Milliy Universiteti, "Universitet" nashriyoti, 2014. - 160 bet.</li> </ol>
<b>Қўшимча адабиётлар:</b>	<ol style="list-style-type: none"> <li>1. Павловская Т.С. Щупак Ю.С. C/C++. Структурное программирование. Практикум.-СПб.: Питер. 2002. -240с</li> <li>2. Павловская Т.С. Щупак Ю.С. С++. Объектно - ориентированное программирование. Практикум.-СПб.: Питер. 2005. -265 с.</li> <li>3. Scheinerman Edwant C++ for Mathematicifns. AnIntroduction for Students and Professionals. Chapman&amp;Hall/CRC,Taylor&amp;Francis Group, LLC, Boca Raton, London, New York, 2006.</li> <li>4. Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.</li> <li>5. Архангельский А.Я. С++ Builder 6. Справочное пособие. Книга 1. Язык С++. -М.: Бином-Пресс. 2002. -544с.</li> <li>6. Культин Н.Б. С++Builder в задачах и примерах.-СПб.: БХВ-Петербург, 2005. -336 с.</li> <li>7. Абрамов С.А., Гнезделова Капустина Е.Н. и др. Задачи по программированию. - М.: Наука, 1988. - 224 с.</li> </ol>
<b>Норматив-ҳуқуқий ҳужжатлар:</b>	<ol style="list-style-type: none"> <li>1. Каримов И.А. Юксак малакали мутахассислар – тараққиёт омили. Т., Ўзбекистон, 1995 й.</li> <li>2. Указ Президента Республики Узбекистан «О дальнейшем развитии компьютеризации и внедрении информационно-коммуникационных технологий»// «Народное слово», 2002 г., 1-июня.</li> <li>3. Постановление Кабинета Министров Республики Узбекистан «О дальнейшем развитии компьютеризации и внедрении информационно-коммуникационных технологий»// «Народное слово», 2002 г., 8-июня.</li> </ol>
<b>Илмий журналлар:</b>	<ol style="list-style-type: none"> <li>1. Science of Computer Programming</li> <li>2. Scientific Programming</li> </ol>

<i>Даврий наирлар:</i>	1. Programming and Computer Software
<i>Статистик наирлар:</i>	-
<i>Интернет ресурслар:</i>	<a href="http://cppstudio.com">http://cppstudio.com</a> <a href="http://cplusplus.com">http://cplusplus.com</a> <a href="http://compteacher.ru/programming">http://compteacher.ru/programming</a> <a href="http://www.intuit.ru/">http://www.intuit.ru/</a> <a href="http://www.ziyonet.uz">http://www.ziyonet.uz</a>



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
3.04	Чизикли алгебра ва аналитик геометрия	246		144	72	72				102	8							
3.05	Геометрия	122		72	36	36				50		4						
3.06	Оддий дифференциал тенгламалар	242		144	72	72			ки	98			4	4				
3.07	Топология	122		72	36	36				50			4					
3.08	Дифференциал геометрия	122		72	36	36				50				4				
3.09	Дискрет математика	182		108	54	54				74			6					
3.10	Математик мантиқ	122		72	36	36				50		4						
3.11	Эхтимоллар назарияси	182		108	54	54				74					4	2		
3.12	Сонли усуллар	122		72	32	40			ки	50					4			
3.13	Хусусий хосилалар дифференциал тенгламалар	244		144	72	72				100					4	4		
3.14	Оптималлаштириш усуллари	142		84	42	42				58							4	4
	Умумий психология	90		54	26			28		36							3	
3.15	Умумий педагогика	90		54	26			28		36					3			
	Танлов фанлари	336		192	68	92		32		144							6	12
<b>4.00</b>	<b>Ихтисослик фанлари</b>	<b>612</b>	<b>9</b>	<b>372</b>	<b>180</b>	<b>164</b>		<b>28</b>	<b>3 ки</b>	<b>240</b>				<b>4</b>	<b>3</b>	<b>5</b>	<b>6</b>	<b>9</b>
4.01	Функционал анализ	270		162	82	80				108				4	3	2		
4.02	Комплекс анализ	158		94	46	48				64						3	4	
4.03	Танлов фанлари	184		116	52	36		28		68							2	9
<b>5.00</b>	<b>Қўшимча тайёргарлик</b>	<b>450</b>	<b>6</b>	<b>216</b>	<b>108</b>	<b>108</b>				<b>234</b>						<b>6</b>	<b>6</b>	
	<b>ЖАМИ</b>	<b>6966</b>		<b>4128</b>	<b>1760</b>	<b>1944</b>		<b>424</b>	<b>3 ки</b>	<b>2838</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>
	<b>Малака амалиёт</b>	<b>1026</b>																
	<b>Битирув малакавий иши</b>	<b>324</b>																
	<b>Аттестациялар</b>	<b>1026</b>																
	<b>ҲАММАСИ</b>	<b>9342</b>																

#### Изоҳ:

- Олий таълим муассасаси ихтисослик фанлари рўйхатини тузишда кадрлар буюртмачиларининг талабларини эътиборга олади.
- Ҳарбий тайёргарлик машгулотлари қўшимча фанлар блокнинг соатлари ҳисобига, ҳарбий йиғин эса таътил вақти ҳисобига ўтказилади. Ҳарбий тайёргарлик машгулотлари ўтказилмайдиган ҳолларда меҳнат бозори ва кадрлар буюртмачиларининг талабларига мосланувчанлиги ва ҳаракатчанлигини таъминлаш учун Илмий Кенгашнинг қарори билан фойдаланилади.
- Ўқув режа асосида олий таълим муассаси ҳар йили ишчи ўқув режасини тузади. Бунда олий таълим муассасасига талабалар юкласининг ҳафталик ҳажмини сақлаган ҳолда ўқув фанлари блоки ҳажмини 5 фоизгача, блоклар таркибидаги фанлар ҳажмини 10 фоизгача ўзгартириш ҳуқуқи берилади.
- Ўқув фанлари ҳажмининг камида 25 фоизи мустақил таълим тарзида ўзлаштирилиши шарт.
- Талаба билимини баҳолаш рейтинг тизимига мувофиқ ўқув жараёни давомида амалга оширилади.
- Битирув малакавий ишини бажариш муддатлари таркибига уни ҳимоя қилиш ҳам киритилади.
- Чет тили фанининг охириги 7-8-семестрларида битирувчи курслар учун қўшимча ва танлов фанлар блоки соатлари ҳисобидан ҳар ҳафтада 2 соатдан “Амалий инглиз тили” фани ўқитилади.
- \* Жисмоний маданият ва спорт фани таркибида «Валеология асослари» назарий курсидан 16 соат ҳажмида маъруза, 12 соат ҳажмида амалий машгулот ўқитилиши кўзда тутилади.
- Курс иши (лойиҳаси) “Умумқасбий фанлар” ва “Ихтисослик фанлар” блокларидаги фанлар бўйича бериш тавсия этилади.
- 1-3-курслардаги малакавий амалиётлар таътил вақтига тўғри келганлиги сабабли ҳар бир курсда мос равишда назарий таълим билан бирга ўтказилади.
- Гуманитар ва ижтимоий-иқтисодий фанлар блокадаги Педагогика. Психология фани умумқасбий фанлар блокадаги Умумий педагогика, Умумий психология фанларига қўшиб ўқитиш режалаштирилган.

Ўқув жараёнининг таркибий қисмлари	Ҳафталар сони	Семестр	Давлат аттестацияси
Назарий таълим	129	1-8	1. Гуманитар ва ижтимоий-иқтисодий фанлардан
Малака амалиёти	18	2, 4, 6, 7	2. Чет тили
Аттестациялар	16+3 (Д)	1-8	3. Битирув малакавий ишини ҳимоя қилиш
Битирув малакавий иши	6	8	
Таътил	32	1-8	
<b>Жами</b>	<b>204</b>		

Мувофиқлаштирувчи кенгаш раиси \_\_\_\_\_

(имзо)

Олий таълим муассасалари бош бошқармаси бошлиғи

**И. Маждидов**

Маънавий ва ахлоқий тарбия бошқармаси бошлиғи

**М. Комилов**

ОЎМКХТРМ директори

**Б. Рахимов**

ЎЗМУ ректори

**Ш. Сирожиддинов**

Ўзбекистон Республикаси Олий ва ўрта махсус таълим вазирлигининг Олий ва ўрта махсус, касб-хунар таълими йўналишлари бўйича ўқув-услубий бирлашмалар фаолиятини Мувофиқлаштирувчи кенгашда маъқулланган

201\_\_ йил “\_\_” \_\_\_\_\_ даги \_\_\_\_\_ - сонли баённома



**II. 2. НАМУНАВИЙ ВА ИШЧИ ЎҚУВ ДАСТУР**  
**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ**  
**ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ**

Рўйхатга олинди  
№ БД – 5130100 – 2.01  
201\_ йил “ \_\_\_ ” \_\_\_\_\_

Олий ва ўрта махсус таълим  
вазирлиги

\_\_\_\_\_

201\_\_ йил “ \_\_\_\_\_ ” \_\_\_\_\_

**ДАСТУРЛАШ АСОСЛАРИ**  
**ФАН ДАСТУРИ**

Билим соҳаси: 300000 – Ишлаб чиқариш-техник соҳа

Таълим соҳаси: 330000 – Компьютер технологиялари ва информатика

Таълим йўналиши: 5130100- Математика

Тошкент – 201\_

Ўзбекистон Республикаси Олий ва ўрта махсус таълим вазирлигининг 201\_\_ йил “\_\_\_” \_\_\_\_\_даги “\_\_\_”-сонли буйруғининг \_\_\_-иловаси билан фан дастури рўйхати тасдиқланган.

Фан дастури Олий ва ўрта махсус, касб-хунар таълими йўналишлари бўйича Ўқув-услугий бирлашмалар фаолиятини Мувофиқлаштирувчи Кенгашининг 201\_\_ йил “\_\_\_” \_\_\_\_\_даги \_\_\_ - сонли баённомаси билан маъқулланган.

Фан дастури Мирзо Улуғбек номидаги Ўзбекистон Миллий университетида ишлаб чиқилди

### **Тузувчилар:**

- Мадрахимов Ш.Ф. - физика-математика фанлари номзоди, ЎзМУ доценти;  
Икрамов А.М. - физика-математика фанлари номзоди, ЎзМУ доценти  
в.б.

### **Такризчилар:**

- Гулямов Ш.М. - техника фанлар доктори, ТДТУ профессори;  
Умаров А. - АТ «Агробанк» тўлов маркази бошлиғи ўринбосари,  
ТАТУ АТ кафедраси доценти, и.ф.н.

Фан дастури Мирзо Улуғбек номидаги Ўзбекистон Миллий университети Услугий кенгашида кўриб чиқилган ва тавсия қилинган (201\_\_ йил “\_\_\_” \_\_\_\_\_даги \_\_\_ - сонли баённома).

## Кириш

“Дастурлаш асослари” фанининг бош мақсади талабаларга қўйилган масалани ечадиган компьютер дастурини тузиш асосларини ўргатишдир. Шу мақсадда дастурлаш тиллари ва муҳитлари ҳақида таянч тушунчалар берилади ва бу тиллардан фойдаланишга ўргатилади.

Фан назарий ва амалий қисмлардан иборат. Назарий қисм информатика ва ҳисоблаш техникаси, алгоритмлар, C/C++ дастурлаш тили, Visual C++ объектга йўналтирилган дастурлаш муҳитларида ишлаш бўйича кўрсатмалар бўлимларидан ташкил топган.

Дастурда компьютерда дастурлашга киришнинг назарий асоси бўлган алгоритмларга алоҳида эътибор қаратилган. Бу ерда алгоритмларни тавсифлаш ва кейинчалик компьютерда амалга ошириш учун зарур бўлган бир катор математик тушунчалар - такрорлаш, ёрдамчи алгоритм, рекурсия, хотира, массив, индекс, функция, параметр ва ҳ.к. киритилиб, турли хил синф масалаларининг алгоритмлари тузилади.

Дастурлаш тили - тузилган алгоритмни компьютер амалга ошириши учун воситадир. Бу ўринда турли мураккабликдаги синтаксис ва семантикага эга бўлган тиллардан фойдаланиш мумкин.

### Фаннинг мақсади ва вазифалари

Фанни ўқитишдан мақсад – “Математика” йўналишининг бакалавр босқичи талабаларига дастурлаш асосларини етарли даражада ўқитиш, шу билимларга таянган ҳолда компьютер ёрдамида моделлаштиришга келадиган тадбиқий масалаларнинг дастур таъминотини амалга оширишга ўргатиш ва ихтисослик фанларини ўзлаштиришда таянч билимларга эга бўлиш.

Фаннинг вазифалари – масала ечишнинг алгоритмик асосларини ўрганиш, компьютер ишлашининг тамоили, дастурлаш тилларини синфлаш, компьютерда берилганлар ва буйруқларни тасвирланиши, C++ тилида дастурлаш, объектга йўналтирилган дастурлаш технологиялари, визуал дастурлаш муҳитида ишлаш бу фаннинг асосий вазифалари ҳисобланади.

### Фан бўйича билим, малака ва кўникмага қўйиладиган талаблар

“Дастурлаш асослари” фанини ўзлаштириш жараёнида амалга ошириладиган масалалар доирасида бакалавр ахборот, уни сақлаш усуллари, қайта ишлаш ва узатиш, ҳисоблаш тизимларининг математик ва дастурий таъминоти, уларни фан соҳаларида, ишлаб чиқариш ва таълимда қўллаш хусусиятлари, компьютерни дастурий таъминоти, дастур турлари ва хусусиятлари, структурали, дастурни оптималлаштириш ва умумлаштириш, дастурлашда модулли тамойилларини қўллаш, компьютер технологиялари ютуқларини замонавий ҳисоблаш тизимларининг математик ва дастурий таъминотида қўллаш, дастурлашнинг тараққиётининг аънаналари ҳақида тасаввурга эга бўлиши, юқори даражадаги дастурлаш тилларини, дастурий таъминотни, дастурлаш технологияларини, тадбиқий ва ҳисоблаш математикаси масалаларини ечиш алгоритмларини, модулли таҳлил ва модулли дастурлаш асосларини, самарали дастур ва дастурлар комплексини

яратиш усулларини билиши ва улардан фойдалана олиши, тадбикий масалаларни ечиш алгоритмини тузиш, математик (компьютер) моделини куриш ва унинг дастурий таъминотини яратиш, структурали, объектга йўналтирилган ва умумлашган дастурлаш парадигмаларини қўллаш асосида иловаларни ярата олиш, дастурлашда, ҳисоблаш техникаси ва дастурий таъминот имкониятларидан самарали фойдаланиш, яратилган иловаларни баҳолаш кўникмаларига эга бўлиши керак.

### **Фаннинг ўқув режадаги бошқа фанлар билан ўзаро боғлиқлиги ва услубий жиҳатдан узвийлиги**

“Дастурлаш асослари” фани йўналишнинг ўқув режасидаги “Эхтимоллар назарияси”, “Сонли усуллар”, “Дискрет математика ва математик мантик” фанлари билан узвий боғлиқ. Фан мазмуни йўналишнинг ўқув режасидаги “Математик статистика”, “Илмий ҳисоблашлар”, “Механика”, “Оддий дифференциал тенгламалар”, “Хусусий ҳосилали дифференциал тенгламалар” фанларини ўзлаштиришда таянч ҳисобланади.

“Дастурлаш асослари” фани умумқасбий фан ҳисобланади ва ўқув йилининг 1-2 семестрларида ўқитилади. Фанни ўқитиш маъруза, амалий машғулот ва мустақил таълим шаклида олиб борилади.

### **Фаннинг ишлаб чиқаришдаги ўрни**

Мазкур дастурга кўра ушбу фан доирасида кўплаб модел масалалар ўрганиладики, бу мазкур фанни чуқур ўрганган ҳар бир бакалавр олган билим ва кўникмаларини ишлаб-чиқаришда, илмий-тадқиқот ишларида, шунингдек, таълим тизимида самарали фойдаланиши имконини беради.

### **Фанни ўқитишда замонавий ахборот ва педагогик технологиялар**

“Дастурлаш асослари” фанини ўқитиш маъруза, амалий машғулотлар ва мустақил таълим кўринишида олиб борилади. Фаннинг мазмуни уни ўқитишда замонавий ахборот технологияларидан, хусусан, компьютер техникасидан фойдаланишни тақоза этади. Шу билан биргаликда фанни ўқитишда илғор ва замонавий усулларидан фойдаланиш, янги инфор­мацион-педагогик технологияларни тадбиқ қилиш катта самара бериши шубҳасиз. Хусусан, фанни ўқитишда дарсликлар, ўқув ва услубий қўлланмалар, электрон дарсликлардан ва мустақил таълим учун масофавий таълим сайтларидан фойдаланиш мақсадга мувофиқ ҳисобланади. Ўқув жараёнида ўқитишнинг интерфаол усуллари, муаммоли ўқитиш технологияси, танқидий фикрлаш ривожланишининг педагогик стратегиялари, шахсий йўналганлик асосидаги педагогик технологиялар, ўқитишни дифференциаллаш, ўқитишнинг индивидуаллаш-тириш технологияси, ўқитишнинг комплекс усуллари (ақлий хужум, баҳс-мунозара усули ва б.) каби педагогик технологиялар ва ўқитиш усулларидан фойдаланилади.

## Асосий қисм

### Фаннинг назарий машғулоти мазмуни

C++ тили синтаксиси ва унинг лексик асоси. C++ тили дастурининг тузилиши ва шакли. Берилганлар турлари. C++ тилининг таянч турлари. Ўзгарувчилар ва ифодалар. Амаллар: инкремент, декремент, sizeof, мантиқий, разрядли, таққослаш. Амалларнинг устунликлари ва бажарилиш йўналишлари. Ҳқиш-ёзиш оқимлари (cin, cout). Операторлар. Шарт операторлари.

Такрорлаш операторлари. Бошқарувни узатиш операторлари. Статик массивлар. Функциялар эълон қилиш ва аниқлаш. main() функцияси. Локал ва глобал ўзгарувчилар. Рекурсив функциялар. Фойдаланувчи томонидан аниқланган берилганлар турлари. Стандарт кутубхона функциялари. Кўрсаткичлар ва адрес олувчи ўзгарувчилар. Динамик массивлар. Функция ва массивлар.

Сатрлар. Сатр устида амаллар. Сатр функциялари. Тузилмалар. Бирлашмалар. Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ҳқиш ва ёзиш функциялари. Файлдан ҳқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари. Динамик тузилмалар.

### Амалий машғулотларини ташкил этиш бўйича кўрсатма ва тавсиялар

Амалий машғулотлар ўтказилишидан мақсад дастурлаш бўйича олинган назарий билимларни амалда мустаҳкамлаш ва турли тоифадаги масалаларни ечишга қўллашдан иборат. Амалий машғулотларни бир қисми аудиторияда доскада ечилиши билан ўтказилса, унинг катта қисми бевосита компьютерда амалга оширилиши керак.

### Амалий машғулотларининг тахминий тавсия этиладиган мавзулари:

1. Санок системалари. Бир санок системасидан иккинчисига ўтиш.
2. Берилганларни компьютер хотирасида тасвирланиши. Кодлаш.
3. Масалани ечиш алгоритмини тузиш ва унинг кўринишлари.
4. Бутун сонли арифметика масалалари.
5. Ичма-ич жойлашган такрорланувчи жараёнлар, итерацион жараёнлар.
6. Кетма-кетликларни тартиблаш, оддий саралаш масалалари.
7. C++ тили синтаксиси. C++ тилида дастур тузилиши.
8. Visual C++ муҳитида ишлаш.
9. Ўзгарувчилар, амаллар, ифодалар билан ишлаш.
10. Ҳқиш-ёзиш оқимларида (cin, cout) киритиш-чиқариш усулларида фойдаланиш.
11. Шарт операторлари билан ишлаш.
12. Такрорлаш операторлари билан ишлаш.
13. Бошқарувни узатиш операторлари билан ишлаш.
14. Статик массивлар билан ишлаш.
15. Функцияларни эълони ва аниқлаш. Оддий функциялар тузиш.

16. Функция параметрлари ва қайтарувчи қийматлари билан ишлаш. Функцияларни қайта юклаш.
17. Рекурсив функциялар билан ишлаш.
18. Фойдаланувчи томонидан аниқланган берилганлар турлари билан ишлаш.
19. Стандарт кутубхона функцияларидан фойдаланиш.
20. Кўрсаткичлар билан ишлаш.
21. Динамик массивлар билан ишлаш.
22. ASCIIZ сатрлар ва улар устида амаллар.
23. string туридаги сатрлар ва улар устида амаллар.
24. Тузилмалар билан ишлаш. Бирлашмалар билан ишлаш.
25. Матн файллар ва улар устида амаллар.
26. Бинар файллар ва улар устида амаллар.
27. Файлдан ўқиш-ёзиш функциялари.
28. Файл кўрсаткичини бошқариш функциялари.
29. Файл структуралар билан ишлаш, саралаш ва қидириш алгоритмлари, ифодаларини ҳисоблаш.
30. Динамик тузилмалар билан ишлаш.

*Изоҳ:* Амалий машғулот соатлари ҳажмларидан келиб чиққан ҳолда ишчи дастурда мазкур мавзулар ичидан амалий машғулот мавзулари шакллантирилади.

### **Мустақил таълимни ташкил этишнинг шакли ва мазмуни**

Талаба мустақил таълимнинг асосий мақсади – ўқитувчининг раҳбарлиги ва назоратида муайян ўқув ишларини мустақил равишда бажариш учун билим ва кўникмаларини шакллантириш ва ривожлантириш.

Мустақил ишларни бажариш жараёнида талабалар қуйидаги ишларни бажарадилар:

- дарслик ва ўқув қўлланмалар асосида фан мавзулари бўйича назарий тайёргарлик кўриш, амалий ва лаборатория машғулотларига тайёрланиш;
- тарқатма материаллар бўйича маърузаларни чуқур ўзлаштириш;
- фан мазмунида кўрсатилмаган дастурлаш тиллари ва муҳитлари билан танишиш ва қиёсий таҳлил қилиш;
- масофавий таълим орқали дастурлаш билан турдош фанлар бўйича ўқув курсларида қатнашиш ва мос сертификатларга эга бўлиш тавсия қилинади.

Талаба мустақил ишини ташкил этишда қуйидаги шакллардан фойдаланади:

- берилган мавзулар бўйича ахборот (реферат) тайёрлаш;
- назарий билимларни амалиётда қўллаш;
- макет, модел ва намуналар яратиш ва ҳ.к.

### **Тавсия этилаётган мустақил ишларнинг мавзулари**

1. C ва C++ тили синтаксислари.
2. C++ ва бошқа тилларда (Pascal, C#) модулли дастурлаш.

3. Стандарт кутубхона. Оқим синфлари.
4. Алгоритмлар кутубхонаси. Сонли ҳисоблаш функцияларининг кутубхонаси.
5. .NET технологияси, C# тилини хусусиятлари.
6. 8,16,32 разрядли процессорлар.
7. Турли муҳитларда яратилган дастур объектларини боғлаш.
8. C++ тили учун консол режимидаги дастурлаш муҳитлари.
9. Динамик структураларни қайта ишлаш алгоритмлари.
10. Саралаш ва излашнинг самарали алгоритмлари.
11. C++ тили стандартлари.
12. C++ тили асосида яратилган тиллар (C# ва Java тиллари).

*Изоҳ:* Мустақил таълим соатлари ҳажмларидан келиб чиққан ҳолда ишчи дастурда мазкур мавзулар ичидан мустақил таълим мавзулари шакллантирилади.

### **Фойдаланиладиган адабиётлар рўйхати**

#### **Асосий адабиётлар**

1. Bjarne Stroustrup. The C++ Programming Language (3th Edition). Addison-Wesley, 1997.
2. D.S. Malik. C++ Programming: From Problem Analysis to Program Design. Fifth Edition. Course Technology, 2011.
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида дастурлаш асослари // Тошкент, ЎзМУ, 2009, 196 бет.
4. Madрахimov Sh.F., Ikramov A.M., Babajanov M.R. C++ tilida programmalash bo'yicha masalalar to'plami. O'quv qo'llanma // Toshkent, O'zbekiston Milliy Universiteti, "Universitet" nashriyoti, 2014. - 160 bet.

#### **Қўшимча адабиётлар**

5. Павловская Т.С. Щупак Ю.С. C/C++. Структурное программирование. Практикум.-СПб.: Питер. 2002. -240с
6. Павловская Т.С. Щупак Ю.С. C++. Объектно - ориентированное программирование. Практикум.-СПб.: Питер. 2005. -265 с.
7. Scheinerman Edwant C++ for Mathematicifns. AnIntroduction for Students and Professionals. Chapman&Hall/CRC,Taylor&Francis Group, LLC, Boca Raton, London, New York, 2006.
8. Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.
9. Архангельский А.Я. C++ Builder 6. Справочное пособие. Книга 1. Язык C++. –М.: Бином-Пресс. 2002. -544с.
- 10.Культин Н.Б. C++Builder в задачах и примерах.-СПб.: БХВ-Петербург, 2005. -336 с.
- 11.Абрамов С.А., Гнезделова Капустина Е.Н. и др. Задачи по программированию. - М.: Наука, 1988. - 224 с.

## Интернет манбаалар

12. <http://cppstudio.com> – C++ тилида программалаш бўйича намуналар изоҳлари билан келтирилган
13. <http://cplusplus.com> – C++ тилида мавжуд конструкциялар таърифи, ишлатиш намуналари билан келтирилган.
14. <http://compteacher.ru/programming> – дастурлаш бўйича видео дарсликлар мавжуд.
15. <http://www.intuit.ru/> – интернет университет, дастурлаш бўйича ёзма ва видео маърузалар ўқиш, тест синовларидан ўтиш ва сертификат олиш имконияти мавжуд.
16. <http://www.ziyonet.uz> – дастурлаш асослари бўйича рефератлар топиш мумкин.



Рўйхатга олинди  
№ В 5130100-2.01  
201\_\_ йил “\_\_” \_\_\_\_\_

**“ТАСДИҚЛАЙМАН”**  
Мирзо Улуғбек номидаги  
Ўзбекистон Миллий  
университети  
ўқув ишлари бўйича проректори

---

**ДАСТУРЛАШ АСОСЛАРИ**  
фанидан  
**ИШЧИ ЎҚУВ ДАСТУР**

Таълим йўналиши 5130100 – Математика

Умумий ўқув соати – 212 соат

Шу жумладан:

Маъруза – 42 соат (1-семестр)  
– 18 соат (2-семестр)

Амалий машғулотлар – 48 соат (1-семестр)  
– 18 соат (2-семестр)

Мустақил таълим соати – 86 соат

**Тошкент-2016**

Фаннинг ишчи ўқув дастури “Информатика ва ахборот технологиялари” (дастурий таъминот) йўналиши ўқув дастури ва ўқув режасига мувофиқ ишлаб чиқилди.

**Тузувчилар:** – ф.-м.ф.н., доцент в.б. **А.М. Икромов**

– ўқитувчи **Қ.Т.Махаров**

**Такризчи:** – ф.-м.ф.н., доцент **А.Т. Хайдаров**

Фаннинг ишчи ўқув дастури “Программалаш ва тармоқ технологиялари” кафедрасининг \_\_\_\_\_ 2016 йилдаги мажлисида муҳокама этилди ва маъқулланди. (Баённома №\_\_)

Кафедра мудирини \_\_\_\_\_ **А.Ишмухамедов**

Фаннинг ишчи ўқув дастури Математика факультети Кенгашининг \_\_\_\_\_ 2016 йилдаги \_\_-сонли йиғилишида муҳокама қилиниб тасдиққа тавсия этилган

Факультет декани \_\_\_\_\_ **А.Худойбердиев**

Мазкур ишчи ўқув дастури Мирзо Улуғбек номидаги Ўзбекистон Миллий университети Ўқув-услугий кенгашининг \_\_\_\_\_ 2016 йилдаги мажлисида тасдиқланган. (Баённома №\_\_)

Ўқув-услугий бошқарма бошлиғи \_\_\_\_\_ **З.Жабборов**

## **Кириш**

“Дастурлаш асослари” фанининг бош мақсади талабаларга қўйилган масалани ечадиган компьютер дастурини тузиш асосларини ўргатишдир. Шу мақсадда дастурлаш тиллари ва муҳитлари ҳақида таянч тушунчалар берилади ва бу тиллардан фойдаланишга ўргатилади.

Фан назарий ва амалий қисмлардан иборат. Назарий қисм информатика ва ҳисоблаш техникаси, алгоритмлар, C/C++ дастурлаш тили, Visual C++ объектга йўналтирилган дастурлаш муҳитларида ишлаш бўйича кўрсатмалар бўлимларидан ташкил топган.

Дастурда компьютерда дастурлашга киришнинг назарий асоси бўлган алгоритмларга алоҳида эътибор қаратилган. Бу ерда алгоритмларни тавсифлаш ва кейинчалик компьютерда амалга ошириш учун зарур бўлган бир қатор математик тушунчалар - такрорлаш, ёрдамчи алгоритм, рекурсия, хотира, массив, индекс, функция, параметр ва ҳ.к. киритилиб, турли хил синф масалаларининг алгоритмлари тузилади.

Дастурлаш тили - тузилган алгоритмни компьютер амалга ошириши учун воситадир. Бу ўринда турли мураккабликдаги синтаксис ва семантикага эга бўлган тиллардан фойдаланиш мумкин.

## **Ўқув фанининг мақсади ва вазифалари**

Фанни ўқитишдан мақсад – “Математика” йўналишининг бакалавр босқичи талабаларига дастурлаш асосларини етарли даражада ўқитиш, шу билимларга таянган ҳолда компьютер ёрдамида моделлаштиришга келадиган тадбиқий масалаларнинг дастур таъминотини амалга оширишга ўргатиш ва ихтисослик фанларини ўзлаштиришда таянч билимларга эга бўлиш.

Фаннинг вазифалари – масала ечишнинг алгоритмик асосларини ўрганиш, компьютер ишлашининг тамоили, дастурлаш тилларини синфлаш, компьютерда берилганлар ва буйруқларни тасвирланиши, C++ тилида дастурлаш, объектга йўналтирилган дастурлаш технологиялари, визуал дастурлаш муҳитида ишлаш бу фаннинг асосий вазифалари ҳисобланади.

## **Фан бўйича талабаларнинг билим, малака ва кўникмаларига қўйиладиган талаблар**

“Дастурлаш асослари” фанини ўзлаштириш жараёнида амалга ошириладиган масалалар доирасида бакалавр ахборот, уни сақлаш усуллари, қайта ишлаш ва узатиш, ҳисоблаш тизимларининг математик ва дастурий таъминоти, уларни фан соҳаларида, ишлаб чиқариш ва таълимда қўллаш хусусиятлари, компьютерни дастурий таъминоти, дастур турлари ва хусусиятлари, структурали, дастурни оптималлаштириш ва умумлаштириш, дастурлашда модулли тамойилларини қўллаш, компьютер технологиялари

ютуқларини замонавий ҳисоблаш тизимларининг математик ва дастурий таъминотида қўллаш, дастурлашнинг тараққиётининг анъаналари ҳақида тасаввурга эга бўлиши, юқори даражадаги дастурлаш тилларини, дастурий таъминотни, дастурлаш технологияларини, татбиқий ва ҳисоблаш математикаси масалаларини ечиш алгоритмларини, модулли таҳлил ва модулли дастурлаш асосларини, самарали дастур ва дастурлар комплексини яратиш усулларини билиши ва улардан фойдалана олиши, тадбиқий масалаларни ечиш алгоритмининг тузиш, математик (компьютер) моделини қуриш ва унинг дастурий таъминотини яратиш, структурали, объектга йўналтирилган ва умумлашган дастурлаш парадигмаларини қўллаш асосида иловаларни ярата олиш, дастурлашда, ҳисоблаш техникаси ва дастурий таъминот имкониятларидан самарали фойдаланиш, яратилган иловаларни баҳолаш кўникмаларига эга бўлиши керак.

### **Фанни ўқитишда замонавий ахборот ва педагогик технологиялар**

“Дастурлаш асослари” фанини ўқитиш маъруза, амалий машғулотлар ва мустақил таълим кўринишида олиб борилади. Фаннинг мазмуни уни ўқитишда замонавий ахборот технологияларидан, хусусан, компьютер техникасидан фойдаланишни тақоза этади. Шу билан биргаликда фанни ўқитишда илғор ва замонавий усулларидан фойдаланиш, янги инфор­мацион-педагогик технологияларни тадбиқ қилиш катта самара бериши шубҳасиз. Хусусан, фанни ўқитишда дарсликлар, ўқув ва услубий қўлланмалар, электрон дарсликлардан ва мустақил таълим учун масофавий таълим сайтларидан фойдаланиш мақсадга мувофиқ ҳисобланади. Ўқув жараёнида ўқитишнинг интерфаол усуллари, муаммоли ўқитиш технологияси, танқидий фикрлаш ривожланишининг педагогик стратегиялари, шахсий йўналганлик асосидаги педагогик технологиялар, ўқитишни дифференциаллаш, ўқитишнинг индивидуаллаш-тириш технологияси, ўқитишнинг комплекс усуллари (ақлий хужум, баҳс-мунозара усули ва б.) каби педагогик технологиялар ва ўқитиш усулларидан фойдаланилади.

### **Фаннинг ўқув режадаги бошқа фанлар билан ўзаро боғлиқлиги ва услубий жиҳатдан узвий кетма-кетлиги**

“Дастурлаш асослари” фани йўналишнинг ўқув режасидаги “Эхтимоллар назарияси”, “Сонли усуллар”, “Дискрет математика ва математик мантик” фанлари билан узвий боғлиқ. Фан мазмуни йўналишнинг ўқув режасидаги “Математик статистика”, “Илмий ҳисоблашлар”, “Механика”, “Оддий дифференциал тенгламалар”, “Хусусий ҳосилали дифференциал тенгламалар” фанларини ўзлаштиришда таянч ҳисобланади.

“Дастурлаш асослари” фани умумқасбий фан ҳисобланади ва ўқув йилининг 1-2 семестрларида ўқитилади. Фанни ўқитиш маъруза, амалий машғулот ва мустақил таълим шаклида олиб борилади.

### Фаннинг ишлаб чиқаришдаги ўрни

Мазкур дастурга кўра ушбу фан доирасида кўплаб модел масалалар ўрганиладики, бу мазкур фанни чуқур ўрганган ҳар бир бакалавр олган билим ва кўникмаларини ишлаб-чиқаришда, илмий-тадқиқот ишларида, шунингдек, таълим тизимида самарали фойдаланиши имконини беради.

### Фан бўйича соатлар тақсимоти

№	Фан мавзулари	Талабанинг ўқув юклараси, соат				
		Ҳаммаси	Аудитория ўқув юклараси			Мустақил таълим
			Жами	жумладан		
				Назарий	Амалий машғулот	
1.	С++ тили синтаксиси ва унинг лексик асоси.	6	4	2	2	2
2.	С++ тили дастурининг тузилиши ва шакли	6	4	2	2	2
3.	Берилганлар турлари. С++ тилининг таянч турлари.	6	4	2	2	2
4.	Ўзгарувчилар ва ифодалар.	6	4	2	2	2
5.	Амаллар: инкремент, декремент, sizeof, мантиқий, разрядли, таққослаш. Амалларнинг устунликлари ва бажарилиш йўналишлари.	10	6	2	4	4
6.	Ўқиш-ёзиш оқимлари (cin, cout).	8	4	2	2	4
7.	Операторлар. Шарт операторлари.	12	8	4	4	4
8.	Такрорлаш операторлари. Бошқарувни узатиш операторлари.	14	8	4	4	6
9.	Статик массивлар.	14	8	4	4	6
10.	Функциялар эълон қилиш ва аниқлаш. main() функцияси. Локал ва глобал ўзгарувчилар.	14	8	4	4	6
11.	Рекурсив функциялар.	8	4	2	2	4

12.	Фойдаланувчи томонидан аниқланган берилганлар турлари.	6	4	2	2	2
13.	Стандарт кутубхона функциялари.	8	4	2	2	4
14.	Кўрсаткичлар ва адрес олувчи ўзгарувчилар.	6	4	2	2	2
15.	Динамик массивлар.	12	6	2	4	6
16.	Сатрлар. Сатр устида амаллар. Сатр функциялари.	12	8	4	4	4
17.	Функция ва массивлар.	10	6	2	4	4
18.	Тузилмалар. Бирлашмалар.	14	8	4	4	6
19.	Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ўқиш ва ёзиш функциялари.	14	8	4	4	6
20.	Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари.	12	8	4	4	4
21.	Динамик тузилмалар.	14	8	4	4	6
	<b>Жами:</b>	<b>212</b>	<b>126</b>	<b>60</b>	<b>66</b>	<b>86</b>

## НАЗАРИЙ МАШҒУЛОТЛАР МАЗМУНИ

### 1-маъруза: C++ тили синтаксиси ва унинг лексик асоси.

Дастурлаш тиллари. C++ тилидаги содда программа. C++ тилида изоҳлар. C++ тили алфавити ва лексемалари.

### 2-маъруза: C++ тили дастурининг тузилиши ва шакли.

Тил синтаксиси. Тил семантикаси. Хужжатлашган идентификаторлар. Дастурни форматлаш.

### 3-маъруза: Берилганлар турлари. C++ тилининг таянч турлари.

Берилганларнинг оддий турлари. Белги тури. Мантикий тур. Ҳақиқий сон тури.

### 4-маъруза: Ўзгарувчилар ва ифодалар.

Таянч арифметик амаллар. Турни бошқа турга келтириш. Константа ўзгармаслар. Ифодаларда ўзгарувчилардан фойдаланиш.

### 5-маъруза: Амаллар: инкремент, декремент, sizeof, мантикий, разрядли, таққослаш.

Амалларнинг устунликлари ва бажарилиш йўналишлари.

Инкремент ва декремент амаллари. sizeof амали. Разрядли мантикий амаллар. Чапга ва ўнгга суриш амаллари. Амалларнинг устунликлари ва бажарилиш йўналишлари.

### **6-маъруза: Ўқиш-ёзиш оқимлари (cin, cout).**

Оқимлар. Ўқиш оқими (cin). Ёзиш оқими (cout).

### **7-8-маъруза: Операторлар. Шарт операторлари.**

Таққослаш амаллари. Мантикий операторлар. Шарт операторлари. if оператори. if...else оператори. ?: шарт амали. switch оператори.

### **9-10-маъруза: Такрорлаш операторлари. Бошқарувни узатиш операторлари.**

Такрорланувчи жараёнлар. for такрорлаш оператори. while такрорлаш оператори. do-while такрорлаш оператори. break оператори. continue оператори. goto оператори ва нишонлар.

### **11-12-маъруза: Статик массивлар.**

Массивлар ҳақида тушунча. Массив элементига мурожаат қилиш. Кўп ўлчамли массивлар. Бир ўлчамли массивларни қайта ишлаш. Бир ўлчамли массивларни инициализация қилиш. Кўп ўлчамли массивларни инициализациялаш.

### **13-14-маъруза: Функциялар эълон қилиш ва аниқлаш. main() функцияси. Локал ва глобал ўзгарувчилар.**

Функциялардан фойдаланиш. Олдиндан аниқланган функциялар. Фойдаланувчи томонидан аниқланадиган функциялар. Келишув бўйича аргументлар. Кўриниш соҳаси. Локал ва глобал ўзгарувчилар. :: амали. Жойлаштириладиган (inline) функциялар. Қайта юкланувчи функциялар.

### **15-маъруза: Рекурсив функциялар.**

Рекурсия тушунчаси. «Ханой минораси» масаласи. Фибоначчи сонларини топиш масаласи.

### **16-маъруза: Фойдаланувчи томонидан аниқланган берилганлар турлари.**

Санаб ўтилувчи тур. Санаб ўтилувчи тур устида амаллар бажариш. Функциялар ва enum турлар. typedef билан ишлаш.

### **17-маъруза: Стандарт кутубхона функциялари.**

сстуре кутубхона файли. Математик функциялар кутубхонаси. Белгилар билан ишлаш функциялари. Турларни ўзгартириш функциялари.

### **18-маъруза: Кўрсаткичлар ва адрес олувчи ўзгарувчилар.**

Кўрсаткичлар. Кўрсаткичга бошланғич қиймат бериш. Кўрсаткич устида амаллар. Адресни олиш амали. Кўрсаткичлар ва адрес олувчи ўзгарувчилар функция параметри сифатида.

### **19-маъруза: Динамик массивлар.**

new оператори. delete оператори. Динамик массивлар билан ишлаш. new оператори ёрдамида динамик массивлар билан.

### **20-21-маъруза: Сатрлар. Сатр устида амаллар. Сатр функциялари.**

ASCIIZ-сатрлар. ASCIIZ-сатрлар узунлигини аниқлаш функциялари. ASCIIZ-сатрларни нусхалаш. ASCIIZ-сатрларни улаш. ASCIIZ-сатрларда излаш функциялари. string туридаги сатрлар. Сатр қисмини бошқа сатрга нусхалаш функцияси. Сатр қисмини ўчириш ва алмаштириш функциялари. Сатр қисмини излаш ва солиштириш функциялари.

### **22-маъруза: Функция ва массивлар.**

Бир ўлчамли массив функция параметри сифатида. Кўп ўлчамли массив функция параметри сифатида. Ўзгарувчан параметрли функциялар.

### **23-24-маъруза: Тузилмалар. Бирлашмалар.**

Структуралар. Структура функция аргументи сифатида. Структуралар массиви. Структураларга кўрсаткич. Бирлашмалар ва улар устида амаллар.

### **25-26-маъруза: Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ўқиш ва ёзиш функциялари.**

Файл тушунчаси. Матн файллар. Бинар файллар. Файл оқимлари билан ишлаш. Файлга форматли ёзиш функциялари. Файлдан форматли ўқиш функциялари.

### **27-28-маъруза: Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари.**

Файлга ёзиш функциялари. Файлдан ўқиш функциялари. Файл кўрсаткичини бошқариш функциялари.



## **29-30-маъруза: Динамик тузилмалар.**

Берилганларнинг динамик тузилмалари. Рўйхатлар. Икки томонлама рўйхатлар. Рўйхатдан излаш. Рўйхатга элемент қўшиш.

## **АМАЛИЙ МАШҒУЛОТЛАР МАЗМУНИ**

### **1-амалий машғулот: C++ тили синтаксиси ва унинг лексик асоси.**

C++ тилидаги содда программа тузиш. C++ тилида изоҳлар билан ишлаш.

### **2-амалий машғулот: C++ тили дастурининг тузилиши ва шакли.**

Хужжатлашган идентификаторлар билан ишлаш. Дастурни форматлаш.

### **3-амалий машғулот: Берилганлар турлари. C++ тилининг таянч турлари.**

Берилганларнинг оддий турлари билан ишлаш. Белги тури билан ишлаш. Мантиқий тур билан ишлаш. Ҳақиқий сон тури билан ишлаш.

### **4-амалий машғулот: Ўзгарувчилар ва ифодалар.**

Таянч арифметик амаллар билан ишлаш. Турни бошқа турга келтириш. Константа ўзгармаслар билан ишлаш. Ифодаларда ўзгарувчилардан фойдаланиш.

### **5-6-амалий машғулот: Амаллар: инкремент, декремент, sizeof, мантиқий, разрядли, таққослаш.**

Инкремент ва декремент амаллари билан ишлаш. sizeof амали билан ишлаш. Разрядли мантиқий амаллар билан ишлаш. Чапга ва ўнгга суриш амаллари билан ишлаш.

### **7-амалий машғулот: Ўқиш-ёзиш оқимлари (cin, cout).**

Ўқиш оқими (cin) ва унинг функциялари. Ёзиш оқими (cout) ва унинг функциялари.

### **8-9-амалий машғулот: Операторлар. Шарт операторлари.**

Таққослаш амаллари билан ишлаш. Мантиқий операторлар билан ишлаш. Шарт операторлари билан ишлаш. if операторни ишлатиш. if...else операторини ишлатиш. ?: шарт амалини ишлатиш. switch оператори билан ишлаш.

### **10-11-амалий машғулот: Такрорлаш операторлари. Бошқарувни узатиш операторлари.**

for такрорлаш оператори билан ишлаш. while такрорлаш оператори билан ишлаш. do-while такрорлаш оператори билан ишлаш. break операторини ишлатиш. continue операторини ишлатиш. goto оператори ва нишонлар билан ишлаш. Ичма-ич такрорланувчи жараёнлар.

### **12-13-амалий машғулот: Статик массивлар.**

Массив элементига мурожаат қилиш. Бир ўлчамли массивлар билан ишлаш. Кўп ўлчамли массивлар билан ишлаш. Бир ўлчамли массивларни қайта ишлаш. Бир ўлчамли массивларни инициализация қилиш. Кўп ўлчамли массивларни инициализациялаш.

### **14-15-амалий машғулот: Функциялар эълон қилиш ва аниқлаш. main() функцияси. Локал ва глобал ўзгарувчилар.**

Функциялардан фойдаланиш. Олдиндан аниқланган функциялар билан ишлаш. Фойдаланувчи томонидан аниқланадиган функциялар билан ишлаш. Келишув бўйича аргументларни ишлатиш. Локал ва глобал ўзгарувчилар билан ишлаш. Жойлаштириладиган (inline) функциялар билан ишлаш. Қайта юкланувчи функцияларни ишлатиш.

### **16-амалий машғулот: Рекурсив функциялар.**

Рекурсияга доир масалалар билан ишлаш. «Ханой минораси» масаласи. Фибоначчи сонларини топиш масаласи.

### **17-амалий машғулот: Фойдаланувчи томонидан аниқланган берилганлар турлари.**

Санаб ўтилувчи тур билан ишлаш. Санаб ўтилувчи тур устида амаллар бажариш. Функциялар ва enum турлар билан ишлаш. typedef билан ишлаш.

### **18-амалий машғулот: Стандарт кутубхона функциялари.**

сстуре кутубхона файли функцияларини ишлатиш. Математик функциялар кутубхонаси билан ишлаш. Белгилар билан ишлаш функциялари. Турларни ўзгартириш функциялари.

### **19-амалий машғулот: Кўрсаткичлар ва адрес олувчи ўзгарувчилар.**

Кўрсаткичлар билан ишлаш. Кўрсаткичга бошланғич қиймат бериш. Кўрсаткич устида амаллар бажариш. Адресни олиш амали билан ишлаш.

Кўрсаткичлар ва адрес олувчи ўзгарувчилар функция параметри сифатида ишлатиш.

**20-21-амалий машғулот: Динамик массивлар.**

new операторини ишлатиш. delete операторини ишлатиш. Динамик массивлар билан ишлаш. new оператори ёрдамида динамик массивлар билан ишлаш.

**22-23-амалий машғулот: Сатрлар. Сатр устида амаллар. Сатр функциялари.**

ASCIIZ-сатрлар билан ишлаш. ASCIIZ-сатрлар узунлигини аниқлаш функцияларини ишлатиш. ASCIIZ-сатрларни нусхалаш. ASCIIZ-сатрларни улаш. ASCIIZ-сатрларда излаш функциялари билан ишлаш. string туридаги сатрлар билан ишлаш. Сатр қисмини бошқа сатрга нусхалаш функциясини ишлатиш. Сатр қисмини ўчириш ва алмаштириш функциялари. Сатр қисмини излаш ва солиштириш функциялари.

**24-25-амалий машғулот: Функция ва массивлар.**

Бир ўлчамли массивни функция параметри сифатида ишлатиш. Кўп ўлчамли массивни функция параметри сифатида ишлатиш. Ўзгарувчан параметрли функциялар билан ишлаш.

**26-27-амалий машғулот: Тузилмалар. Бирлашмалар.**

Структуралар билан ишлаш. Структура функция аргументи сифатида ишлатиш. Структуралар массиви билан ишлаш. Бирлашмалар ва улар устида амаллар

**28-29-амалий машғулот: Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ўқиш ва ёзиш функциялари.**

Матн файллар билан ишлаш. Бинар файллар билан ишлаш. Файл оқимлари билан ишлаш. Файлга форматли ёзиш функциялари. Файлдан форматли ўқиш функциялари.

**30-31-амалий машғулот: Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари.**

Файлга ёзиш функциялари. Файлдан ўқиш функциялари. Файл кўрсаткичини бошқариш функциялари.

### 32-33-амалий машғулот: Динамик тузилмалар.

Берилганларнинг динамик тузилмалари билан ишлаш. Рўйхатлар билан ишлаш. Икки томонлама рўйхатлар билан ишлаш. Рўйхатдан излаш. Рўйхатга элемент қўшиш.

#### Баҳолаш мезони

##### Ўзлаштириш назорати (1-семестр)

№	Рейтинг назорат / шакли, максимал баллари	1-ЖН	2-ЖН	3-ЖН (МТ)	1-ОН	2-ОН	ЯН	Баллар йиғиндиси
1.	Максимал балл	10	15	15	15	15	30	100
2.	Шакли:	Оғзаки	Оғзаки	Оғзаки	ёзма	ёзма	ёзма	

##### Ўзлаштириш назорати (2-семестр)

№	Рейтинг назорат / шакли, максимал баллари	1-ЖН	2-ЖН (МТ)	1-ОН	ЯН	Баллар йиғиндиси
1.	Максимал балл	20	30	20	30	100
2.	Шакли:	Оғзаки	Оғзаки	Ёзма	Ёзма	

#### Фойдаланиладиган адабиётлар рўйхати

##### Асосий адабиётлар

1. Bjarne Stroustrup. The C++ Programming Language (3th Edition). Addison-Wesley, 1997.
2. D.S. Malik. C++ Programming: From Problem Analysis to Program Design. Fifth Edition. Course Technology, 2011.
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида дастурлаш асослари// Тошкент, ЎЗМУ, 2009, 196 бет.

4. Madraximov Sh.F., Ikramov A.M., Babajanov M.R. C++ tilida programmalash bo'yicha masalalar to'plami. O'quv qo'llanma // Toshkent, O'zbekiston Milliy Universiteti, "Universitet" nashriyoti, 2014. - 160 bet.
5. Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005. 1182 page

### **Қўшимча адабиётлар**

6. Bjarne Stroustrup. The C++ Programming Language (4th Edition). Addison-Wesley, 2013. 1363 page.
7. Bjarne Stroustrup. Programming: Principles and Practice using C++ (Second Edition)" Addison-Wesley, 2014, 1305 page.
8. Павловская Т.А. C++. Программирование на языке высокого уровня – СПб.: Питер. 2005.- 461 с.
9. Walter Savitch. Absolute C++, 5th edition. Addison-Wesley/Pearson, 2012. 984 page.
10. Walter Savitch. Problem Solving with C++, 9th edition. Addison-Wesley/Pearson, 2015. 1088 page.
11. Павловская Т.С. Щупак Ю.С. C/C++. Структурное программирование. Практикум.-СПб.: Питер, 2002.-240с
12. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования C++: Учебный курс.- Харьков: Фолио; М.: ООО «Издательство АСТ», 2001.-500с.
13. Культин Н.Б. C++Builder в задачах и примерах.-СПб.: БХВ-Петербург, 2005.-336с.
14. Абрамов С.А., Гнезделова Капустина Е.Н. и др. Задачи по программированию. - М.: Наука, 1988.

### **Интернет манбаалар**

15. <http://cppstudio.com> – C++ тилида программалаш бўйича намуналар изохлари билан келтирилган
16. <http://cplusplus.com> – C++ тилида мавжуд конструкциялар таърифи, ишлатиш намуналари билан келтирилган.
17. <http://compteacher.ru/programming> – дастурлаш бўйича видео дарсликлар мавжуд.
18. <http://www.intuit.ru/> – интернет университет, дастурлаш бўйича ёзма ва видео маърузалар ўқиш, тест синовларидан ўтиш ва сертификат олиш имконияти мавжуд.
19. <http://www.ziyonet.uz> – дастурлаш асослари бўйича рефератлар топиш мумкин.

### III. МОДУЛНИ ЎҚИТИШДА ФОЙДАЛАНИЛАДИГАН ИНТРЕФАОЛ ТАЪЛИМ МЕТОДЛАРИ

#### 1. “Кейс-стади” методи

«Кейс-стади» - инглизча сўз бўлиб, («case» – аниқ вазият, ҳодиса, «stadi» – ўрганмоқ, таҳлил қилмоқ) аниқ вазиятларни ўрганиш, таҳлил қилиш асосида ўқитишни амалга оширишга қаратилган метод ҳисобланади. Мазкур метод дастлаб 1921 йил Гарвард университетиде амалий вазиятлардан иқтисодий бошқарув фанларини ўрганишда фойдаланиш тартибида қўлланилган. Кейсда очиқ ахборотлардан ёки аниқ воқеа-ҳодисадан вазият сифатида таҳлил учун фойдаланиш мумкин. Кейс ҳаракатлари ўз ичига қуйидагиларни қамраб олади: Ким (Who), Қачон (When), Қерда (Where), Нима учун (Why), Қандай/ Қанақа (How), Нима-натижа (What).

#### “Кейс методи” ни амалга ошириш босқичлари

Иш босқичлари	Фаолият шакли ва мазмуни
<b>1-босқич:</b> Кейс ва унинг ахборот таъминоти билан таништириш	<ul style="list-style-type: none"> <li>✓ якка тартибдаги аудио-визуал иш;</li> <li>✓ кейс билан танишиш(матнли, аудио ёки медиа шаклда);</li> <li>✓ ахборотни умумлаштириш;</li> <li>✓ ахборот таҳлили;</li> <li>✓ муаммоларни аниқлаш</li> </ul>
<b>2-босқич:</b> Кейсни аниқлаштириш ва ўқув топшириғни белгилаш	<ul style="list-style-type: none"> <li>✓ индивидуал ва гуруҳда ишлаш;</li> <li>✓ муаммоларни долзарблик иерархиясини аниқлаш;</li> <li>✓ асосий муаммоли вазиятни белгилаш</li> </ul>
<b>3-босқич:</b> Кейсдаги асосий муаммони таҳлил этиш орқали ўқув топшириғининг ечимини излаш, ҳал этиш йўлларини ишлаб чиқиш	<ul style="list-style-type: none"> <li>✓ индивидуал ва гуруҳда ишлаш;</li> <li>✓ муқобил ечим йўлларини ишлаб чиқиш;</li> <li>✓ ҳар бир ечимнинг имкониятлари ва тўсиқларни таҳлил қилиш;</li> <li>✓ муқобил ечимларни танлаш</li> </ul>
<b>4-босқич:</b> Кейс ечимини ечимини шакллантириш ва асослаш, тақдимот.	<ul style="list-style-type: none"> <li>✓ якка ва гуруҳда ишлаш;</li> <li>✓ муқобил вариантларни амалда қўллаш имкониятларини асослаш;</li> <li>✓ ижодий-лойиха тақдимотини тайёрлаш;</li> <li>✓ якуний хулоса ва вазият ечимининг амалий аспектларини ёритиш</li> </ul>

**Кейс.** Динамик массивлар билан ишлайдиган дастур тузилди. Дастур вазифаси массив элементларини циклик равишда чапга n та суриш. Дастур ишлаши натижасида хатолик келиб чиқди. Яъни илова хатолик ҳақида хабар берди.

## **Кейсни бажариш босқчилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

## **2. “Ассесмент” методи**

**Методнинг мақсади:** мазкур метод таълим олувчиларнинг билим даражасини баҳолаш, назорат қилиш, ўзлаштириш кўрсаткичи ва амалий кўникмаларини текширишга йўналтирилган. Мазкур техника орқали таълим олувчиларнинг билиш фаолияти турли йўналишлар (тест, амалий кўникмалар, муаммоли вазиятлар машқи, қиёсий таҳлил, симптомларни аниқлаш) бўйича ташҳис қилинади ва баҳоланади.

### **Методни амалга ошириш тартиби:**

“Ассесмент” лардан маъруза машғулотида талабаларнинг ёки катнашчиларнинг мавжуд билим даражасини ўрганишда, янги маълумотларни баён қилишда, семинар, амалий машғулотларда эса мавзу ёки маълумотларни ўзлаштириш даражасини баҳолаш, шунингдек, ўз-ўзини баҳолаш мақсадида индивидуал шаклда фойдаланиш тавсия этилади. Шунингдек, ўқитувчининг ижодий ёндашуви ҳамда ўқув мақсадларидан келиб чиқиб, ассесментга кўшимча топшириқларни киритиш мумкин.

**Намуна.** Ҳар бир катакдаги тўғри жавоб 5 балл ёки 1-5 балгача баҳоланиши мумкин.

### **Тест**

- 1.C++ тилида динамик массив яратиш учун қайси калит сўз ишлатилади?
- A. new
- B. delete
- C. float

### **Қиёсий таҳлил**

- Статик ва динамик массивларни таҳлил қилинг?

### **Тушунча таҳлили**

- Динамик сўзини изоҳланг...

### **Амалий кўникма**

- Динамик массив яратиш усулларини ўрганинг.

### 3. “Тушунчалар таҳлили” методи

**Методнинг мақсади:** мазкур метод талабалар ёки қатнашчиларни мавзу буйича таянч тушунчаларни ўзлаштириш даражасини аниқлаш, ўз билимларини мустақил равишда текшириш, баҳолаш, шунингдек, янги мавзу буйича дастлабки билимлар даражасини ташҳис қилиш мақсадида қўлланилади. Методни амалга ошириш тартиби:

- иштирокчилар машғулот қоидалари билан таништирилади;
- ўқувчиларга мавзуга ёки бобга тегишли бўлган сўзлар, тушунчалар номи туширилган тарқатмалар берилади (индивидуал ёки гуруҳли тартибда);
- ўқувчилар мазкур тушунчалар қандай маъно англатиши, қачон, қандай ҳолатларда қўлланилиши ҳақида ёзма маълумот берадилар;
- белгиланган вақт якунига етгач ўқитувчи берилган тушунчаларнинг тугри ва тулиқ изоҳини уқиб эшиттиради ёки слайд орқали намойиш этади;
- ҳар бир иштирокчи берилган тугри жавоблар билан узининг шахсий муносабатини таққослайди, фарқларини аниқлайди ва ўз билим даражасини текшириб, баҳолайди.

**Намуна:** “Модулдаги таянч тушунчалар таҳлили”

Тушунчалар	Сизнингча бу тушунча қандай маънони англатади?	Қўшимча маълумот
Идентификатор	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизиқ (‘_’) белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги	
new	хотирадан янги жой ажратиш оператори	
delete	хотирадан ажратилган жойни тозалаш оператори	
Кўрсаткич	қийматлари адрес бўлган ўзгарувчилар	
Адрес	ўзгарувчи хотирада жойлашадиган манзил	
cout	экранга чиқариш оқими.	
cin	экрандан киритиш оқими	
Ўзгарувчи	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги	

**Изоҳ:** Иккинчи устунчага қатнашчилар томонидан фикр билдирилади. Мазкур тушунчалар ҳақида қўшимча маълумот глоссарийда келтирилган.



# I CEMENTP

## IV. МАЪРУЗА МАТЕРИАЛЛАРИ

### Маъруза 1. C++ тили синтаксиси ва унинг лексик асоси

#### Режа:

1. Дастурлаш тиллари
2. C++ тилидаги содда программа
3. C++ тилида изоҳлар
4. C++ тили алфавити ва лексемалари

Таянч иборалар: *Дастурлаш тиллари, include, cout, кенгайтма, Компиляция, Лексема.*

#### 1.1. Дастурлаш тиллари

Маълумки, дастур машина кодларининг қандайдир кетма-кетлиги бўлиб, аниқ бир ҳисоблаш воситасини амал қилишини бошқаради. Дастур таъминотини яратиш жараёнини осонлаштириш учун юзлаб дастурлаш тиллари яратилган. Барча дастурлаш тилларини икки тоифага ажратиш мумкин:

- куйи даражадаги дастурлаш тиллари;
- юқори даражадаги дастурлаш тиллари.

Куйи даражадаги дастурлаш тилларига Assembler туридаги тиллар киради. Бу тиллар нисбатан қисқа ва тезкор бажарилувчи кодларни яратиш имкониятини беради. Лекин, Assembler тилида дастур тузиш заҳматли, нисбатан узоқ давом этадиган жараёндир. Бунга қарама-қарши равишда юқори босқич тиллари яратилганки, уларда табиий тилнинг чекланган кўринишидан фойдаланган ҳолда дастур тузилади. Юқори босқич тилларидаги операторлар, берилганларнинг турлари, ўзгарувчилар ва дастур ёзишнинг турли усуллари тилнинг ифодалаш имконияти оширади ва дастурни «ўқимишли» бўлишини таъминлайди. Юқори босқич тилларига Fortran, PL/1, Prolog, Lisp, Basic, Pascal, C ва бошқа тилларни мисол келтириш мумкин. Компютер архитектурасини такомиллашуви, компютер тармоғининг ривожланиши мос равишда юқори босқич тилларини янги вариантларини юзага келишига, янги

тилларни пайдо бўлишига, айрим тилларни эса йўқолиб кетишига олиб келди. Ҳозирда кенг тарқалган тилларга Object Pascal, C++, C#, Php, Java тиллари ҳисобланади. Хусусан, C тилининг такоммиллашган варианты сифатида C++ тилини олишимиз мумкин. 1972 йилда Денис Ритч ва Браян Кернеги томонидан C тили яратилди. 1980 йилда Бьярн Страустроп C тилининг авлоди C++ тилини яратдики, унда структурали ва объектга йўналтирилган дастурлаш технологиясига таянган ҳолда дастур яратиш имконияти туғилди.

## 1.2. C++ тилидаги содда программа

Қуйида C++ тилидаги содда программа матни келтирилган.

```
#include <iostream>           // сарлавҳа файлни кўшиш
using namespace std;         // std номлар фазосини ишлатиш
int main ()                  // бош функция тавсифи
{                             // блок бошланиши
    cout << "Salom Olam! \n"; // сатрни чоп этиш
    return 0;                // функция қайтарадиган қиймат
}                             // блок тугаши
```

Программа бажарилиши натижасида экранга "Salom Olam!" сатри чоп этилади.

Программанинг 1-сатрида #include – препроцессор директиваси бўлиб, программа кодига оқимли ўқиш/ёзиш функциялари ва унинг ўзгарувчилари эълони жойлашган «iostream» сарлавҳа файлини кўшади. Кейинги қаторларда программанинг ягона, асосий функцияси - main() функцияси тавсифи келтирилган. Шунини қайд этиш керакки, C++ программасида албатта main() функцияси бўлиши шарт ва программа шу функцияни бажариш билан ўз ишини бошлайди.

C++ тилида программа яратиш бир нечта босқичлардан иборат бўлади. Дастлаб, матн таҳририда (одатда программалаш муҳитининг таҳририда) программа матни терилади, бу файлни кенгайтмаси «.cpp» бўлади. Кейинги босқичда программа матни ёзилган файл компиляторга узатилади, агарда программада хатоликлар бўлмаса, компилятор «.obj» кенгайтмали объект модул файлини ҳосил қилади. Охириги қадамда компоновка (йиғувчи) ёрдамида «.exe» кенгайтмали бажарилувчи файл программа ҳосил бўлади.

Босқичларда юзага келувчи файлларнинг номлари бошланғич матн файлининг номи билан бир хил бўлади.

Компиляция жараёнининг ўзи ҳам иккита босқичдан ташкил топади. Бошида препроцессор ишлайди, у матндаги компиляция директиваларини бажаради, хусусан `#include` директиваси бўйича кўрсатилган кутубхоналардан C++ тилида ёзилган модулларни программа таркибига киритади. Шундан сўнг кенгайтирилган программа матни компиляторга узатилади. Компилятор ўзи ҳам программа бўлиб, унинг учун кирувчи маълумот бўлиб, C++ тилида ёзилган программа матни ҳисобланади. Компилятор программа матнини лексема (атомар) элементларга ажратади ва уни лексик, кейинчалик синтактик таҳлил қилади. Лексик таҳлил жараёнида у матнни лексемаларга ажратиш учун «пробел ажратувчисини» ишлатади. Пробел ажратувчисига - пробел белгиси (' '), '\t' - табуляция белгиси, '\n'- кейинги қаторга ўтиш белгиси, бошқа ажратувчилар ва изоҳлар (комментарийлар) киради.

### 1.3. C++ тилида изоҳлар

Программа матни тушунарли бўлиши учун изоҳлар ишлатилади. Изоҳлар компилятор томонидан «ўтказиб» юборилади ва улар программа амал қилишига ҳеч қандай таъсир қилмайди.

C++ тилида изоҳлар икки кўринишда ёзилиши мумкин.

Биринчисида `/*` дан бошланиб, `*/` белгалари билан тугаган барча белгилар кетма-кетлиги изоҳ ҳисобланади, иккинчиси «сатрий изоҳ» деб номланади ва у `///  
//` белгилардан бошланган ва сатр охиригача ёзилган белгилар кетма-кетлиги бўлади. Изоҳнинг биринчи кўринишида ёзилган изоҳлар бир неча сатр бўлиши ва улардан кейин C++ операторлари давом этиши мумкин.

Мисол.

```
int main()  
{  
  // бу қатор изоҳ ҳисобланади  
  int a=0; //int d;  
  int c;  
  /* int b=15 */  
  /*- изоҳ бошланиши  
  a=c;
```

```

изоҳ тугаши */
return 0;
}

```

Программада d, b ўзгарувчилар эълонлари инобатга олинмайди ва a=c амали бажарилмайди.

Программа танасида консол режимида белгилар кетма-кетлигини оқимга чиқариш амали қўлланилган. Маълумотларни стандарт оқимга (экранга) чиқариш учун қуйидаги формат ишлатилган:

```
cout << <ифода>;
```

Бу ерда <ифода> сифатида ўзгарувчи ёки синтаксиси тўғри ёзилган ва қандайдир қиймат қабул қилувчи тил ифодаси келиши мумкин (кейинчалик, бурчак қавс ичига олинган ўзбекча сатр остини тил таркибига кирмайдиган тушунча деб қабул қилиш керак).

Масалан:

```
int uzg=324;
cout << uzg;           // бутун сон чоп этилади

```

Берилганларни стандарт оқимдан (одатда клавиатурадан) ўқиш қуйидаги форматда амалга оширилади:

```
cin >> <ўзгарувчи>;
```

Бу ерда <ўзгарувчи> қиймат қабул қилувчи ўзгарувчининг номи.

Мисол:

```
int Yosh;
cout << "Yoshingizni kiriting: ";
cin >> Yosh;

```

Бутун турдаги Yosh ўзгарувчиси киритилган қийматни ўзлаштиради. Киритилган қийматни ўзгарувчи турига мос келишини текшириш масъулияти программа тузувчисининг зиммасига юкланади.

Бир пайтнинг ўзида пробел воситасида бир нечта ва ҳар хил турдаги қийматларни оқимдан киритиш мумкин. Қиймат киритиш <Enter> тугмасини босиш билан тугайди. Агар киритилган қийматлар сони ўзгарувчилар сонидан кўп бўлса, «ортиқча» қийматлар буфер хотирада сақланиб қолади.

```

#include <iostream>
using namespace std;
int main  ()
{

```

```

int x,y;
float z;
cin >> x >> y >> z;
cout << "O'qilgan qiymatlar:\n";
cout << x << '\t' << y << '\t' << z;
return 0;
}

```

Ўзгарувчиларга қиймат киритиш учун клавиатура орқали

10 20 3.14 <Enter>

ҳаракати амалга оширилади. Шуни қайд этиш керакки, оқимга қиймат киритишда пробел ажратувчи ҳисобланади. Ҳақиқий соннинг бутун ва каср қисмлари ' .' белгиси билан ажратилади.

#### 1.4. C++ тили алфавити ва лексемалари

C++ тили алфавити ва лексемаларига қуйидагилар киради:

- катта ва кичик лотин алфавити ҳарфлари;
- рақамлар - 0,1,2,3,4,5,6,7,8,9;
- махсус белгилар: “ { } | [ ] ( ) + - / % \ ; ‘ : ? < = > \_ ! & ~ # ^ . \* ”

Алфавит белгиларидан тилининг лексемалари шакллантирилади: идентификаторлар; калит (хизматчи ёки захираланган) сўзлар; ўзгармаслар; амаллар белгиланишлари; ажратувчилар.

Дастурлаш тилининг муҳим таянч тушунчаларидан бири - идентификатор тушунчасидир. Идентификатор деганда катта ва кичик лотин ҳарфлари, рақамлар ва таг чизиқ (‘\_’) белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги тушунилади. Идентификаторларда ҳарфларнинг регистрлари (катта ёки кичиклиги) ҳисобга олинади. Масалан, RUN, run, Run – бу ҳар хил идентификаторлардир. Идентификатор узунлигига чегара қўйилмаган.

Идентификаторлар калит сўзлар, ўзгарувчилар, функциялар, нишонлар ва бошқа объектларни номлашда ишлатилади.

C++ тилининг калит сўзларига қуйидагилар киради:

asm, auto, break, case, catch, char, class, const, continue, default, delete, do, double, else, enum, explicit, extern, float, for, friend, goto, if, inline, int, long, mutable, new, operator, private, protected, public, register, return, short, signed,

sizeof, static, struct, swith, template, this, throw, try, typedef, typename, union, unsigned, virtual, void, volatile, while.

Юқорида келтирилган идентификаторларни бошқа мақсадда ишлатиш мумкин эмас.

Процессор регистрларини белгилаш учун қуйидаги сўзлар ишлатилади:

`_AH, _AL, _AX, _EAX, _BH, _BL, _BX, _EBX, _CL, _CH, _CX, _ECX, _DH, _DL, _DX, _EDX, _CS, _ESP, _EBP, _FS, _GS, _DI, _EDI, _SI, _ESI, _BP, _SP, _DS, _ES, _SS, _FLAGS.`

Булардан ташқари “`__`” (иккита тагчизик) белгиларидан бошланган идентификаторлар кутубхоналар учун захираланган. Шу сабабли ‘`_`’ ва “`__`” белгиларни идентификаторнинг биринчи белгиси сифатида ишлатмаган маъқул. Идентификатор белгилар орасида пробел ишлатиш мумкин эмас, зарур бўлганда унинг ўрнига ‘`_`’ ишлатиш мумкин: `Silindr_radiusi`, `aylana_diametri`.

```
#include <iostream>
using namespace std;
int main()
{
int son;
son = 6;
cout << "Mening birinchi C++ dasturim." << endl;
cout << " 2 va 3 ning yig'indisi = " << 5 << endl;
cout << "7 + 8 = " << 7 + 8 << endl;
cout << "Son = " << son << endl;
return 0;
}
```

Ушбу дастурни компиляция қилиб ишлатганда экранга қуйидаги тўртта қатор чиқади:

```
Mening birinchi C++ dasturim.
2 va 3 ning yig'indisi = 5
7 + 8 = 15
Son = 6
```

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим мавзулари

1. Дастурлаш тиллари тарихи.
2. C ва C++ дастурлаш тили.
3. Лексемалар ва уларнинг қўлланилиши.
4. Синтаксиснинг дастурлаш тили учун аҳамияти.

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
Дастурлаш тиллари	programming languages	дастур таъминотини яратиш жараёнини осонлаштириш учун яратилган тиллар
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
cout		экранга чиқариш оқими
кенгайтма	extension	файлларнинг турли дастурларга тегишлилигини аниқловчи файл кўринишининг қисми
компиляция	compilation	бажарилувчи файл хосил бўлиш жараёни
Лексема	lexeme	тилнинг ажралмайдиган қисмлари



Идентификатор	identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизик (‘_’) белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги
---------------	------------	-----------------------------------------------------------------------------------------------------------------------------------------

### Кейслар банки

**Кейс 1.** Дастур алгоритми тузилди, ушбу дастур турли бурчакларда котангенс қийматларини ҳисоблаб беради. Лекин дастур ишлаши давомида хатолик юз берди. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

### Назорат саволлари

1. Қуйи даражадаги дастурлаш тиллари нима?
2. Компиляция жараёнининг неча босқичдан ташкил топади?
3. Қандай белгилар билан тугаган барча белгилар кетма-кетлиги изоҳ ҳисобланади?
4. C++ тилининг калит сўзларига қайсилар киради?
5. Процессор регистрларини белгилаш учун қайси сўзлар ишлатилади?
6. Компиляция жараёни нима?
7. Киритилган қийматни ўзгарувчи турига мос келишини қандай текшириш мумкин?
8. Идентификатор нима?
9. “\_\_” (иккита тагчизик) белгиларидан бошланган идентификаторлар нима учун ишлатилади?
10. RUN, run, Run – идентификаторларнинг фарқи нимада?

## Тест саволлари

1. C++ тилида консоль режимда ишлаш жараёнида маълумотларни экранга чиқариш формати тўғри кўрсатилган жавобни топинг.

- a) `cout<<<ifoda>`
- b) `cin>><o'zgaruvchi>`
- c) `cout>><ifoda>`
- d) To'g'ri javob keltirilmagan

2. C++ тилида консоль режимда ишлаш жараёнида берилганларни ўқиш формати тўғри кўрсатилган жавобни аниқланг.

- a) `cin>><o'zgaruvchi>`
- b) `cout<<<ifoda>`
- c) `cin<<<o'zgaruvchi>`
- d) `cout>><o'zgaruvchi>`

3. C++ тилида ҳақакий соннинг бутун ва каср қисмлари қайси белги ёрдамида ажратилади?

- a) Nuqta “.”
- b) Vergul “;”
- c) Nuqta vergul “;”
- d) Slash “/”

4. Идентификатор нима?

- a) катта ва кичик лотин ҳарфлари, рақамлар ва таг чизик (‘\_’) белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги
- b) катта ва кичик лотин ҳарфлари, рақамлар ва таг чизик (‘\_’) белгиларидан ташкил топган ва фақат рақамдан бошланувчи белгилар кетма-кетлиги
- c) катта ва кичик лотин ҳарфларидан ташкил топган кетма-кетлик
- d) катта ва кичик лотин ҳарфлари, рақамлар ва таг чизик (‘\_’) белгиларидан ташкил топган ва таг чизик (‘\_’) билан бошланмайдиган белгилар кетма-кетлиги

5. Блок бу –

- a) '{' ва '}' белги оралиғига олинган операторлар кетма-кетлиги, у компилятор
- b) '(' ва ')' белги оралиғига олинган операторлар кетма-кетлиги, у компилятор томонидан яхлит ьир оператор деб қабул қилинади
- c) '/\*' ва '\*/' белги оралиғига олинган операторлар кетма-кетлиги, у компилятор томонидан яхлит ьир оператор деб қабул қилинади
- d) '[' ва ']' белги оралиғига олинган операторлар кетма-кетлиги, у компилятор томонидан яхлит ьир оператор деб қабул қилинади

6. Қуйидаги программа қисми нима натижа чиқаради?

```
float a=1234; cout<<a/100;
```

- a) 12.34
- b) 12
- c) 34
- d) 1.234

7. Қуйидаги программа қисми нима натижа чиқаради?

```
int a=6789; cout<<a/100;
```

- a) 67
- b) 6789
- c) 89
- d) 678

8. Қуйидаги программа қисми нима натижа чиқаради?

- a) 46
- b) 68
- c) 468
- d) 6

9. Идентификатор сифатида фойдаланиш мумкин бўлмаган жавобни кўрсатинг.

- a) Int
- b) a
- c) A7
- d) Pi

10. Идентификатор сифатида фойдаланиш мумкин бўлмаган жавобни кўрсатинг.

- a) float
- b) abc
- c) A3
- d) kiCHik

## Маъруза 2. C++ тили дастурининг тузилиши ва шакли

### Режа:

1. Тил синтаксиси
2. Тил семантикаси
3. Хужжатлашган идентификаторлар
4. Дастурни форматлаш

Таянч иборалар: *Синтактик қоидалар, include, cout, семантика, Компиляция, Лексема, Идентификатор.*

### 2.1. Тил синтаксиси

Аввалги мавзуда маънога эга дастур тузиш учун керакли бўлган C++ тушунчалари ўрганилди. Дастур тузиш учун аввал унинг ўзига хос структурасини аниқлаш керак. Ўзига хос структурадан фойдаланишдан асосий мақсад – C++ да тузилган дастурни тушунишни осонлаштириш ва дастурни ўзгартиришдан иборат. Синтактик тўғри ёзилган, лекин ҳеч қандай структурага эга бўлмаган дастурни тушуниш ва ўзгартириш жуда қийин ҳамда кўп ресурс, вақт талаб қилади. Шунинг билан бирга, ҳар бир C++ тилида тузилган дастур тилдаги аниқланган қоидаларни қониқтириши зарур. C++ дастурида асосий функция – main() функцияси мавжуд бўлиши керак. Дастур грамматик қоидаларга ўхшаш бўлган синтаксис қоидаларига, яъни тилда нима тўғри, нима нотўғри, нима мумкин, кима мумкин эмаслигини аниқловчи қоидаларга амал қилиш керак. Шунингдек, мақсадга эришишда дастурлаш тилининг маъносини берувчи тилнинг семантик қоидаларига ҳам амал қилиш зарур. Бунда, синтаксис, пробелларнинг ишлатилиши, нукта, вергул, нуктали вергул ва қавсларнинг ишлатилиши ва маъноси, семантика, идентификаторлар ва уларнинг номланиши, қаторларнинг ишлатилиши, изоҳлар ва уларда ишлатилган хужжатлаш, шунингдек, дастурнинг тузилиш шакли ва ёзилиш стилига алоҳида эътибор бериш керак.

Синтаксис – қандай ёзиш мумкин ва қандай ёзиш мумкин эмаслигини аниқлаб берувчи тилнинг қоидалари. Хатолар компиляция жараёнида аниқланади ва дастурчига кўрсатилади.

Маълумки, дастур матни матн мухаррири ёрдамида компьютерга киритилади. Мукамал дастур матнини тузиш жуда қийин, турли кўринишдаги ва қийинчиликдаги хатоликлар бўлиши табиий. Шунинг учун, компиляция жараёнидан кейин юзага келган хатоликлар матн мухаррири томонидан дастурчига тақдим этилади. Шундай хатоликлар бўлиши мумкинки, маълум бир специфик код қисмидаги хатолик дастур кодининг бошқа қисмларида хатоликни юзага келтираётган бўлиши мумкин. Бундай ҳолатларда асосий хатолик бўлган қисмини тўғирлаб, дастурни компиляция қилгандан сўнг қолган хатоликлар тўлиқ ёки қисман йўқолиши мумкин. Шунинг учун, синтаксис хатоларини компилятор кўрсатган кетма-кетликда бартараф этиш мақсадга мувофиқ. Шунингдек, компилятор фақатгина хатоларни аниқлабгина қолмасдан, уларнинг дастур матнининг қайси қисмидалигини ва уни бартараф этиш йўллари хам кўрсатиб беради.

Масалан:

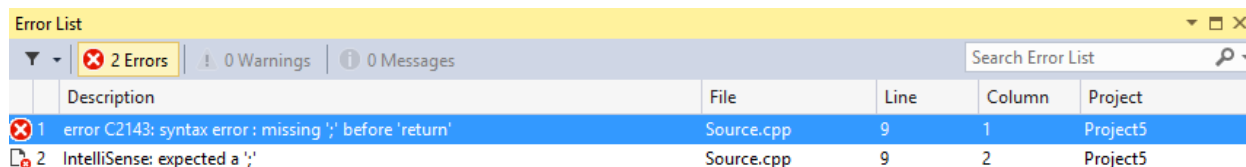
```
#include <iostream>           //1-qator
using namespace std;         //2-qator
int main()                   //3-qator
{                             //4-qator
    int x, y;                 //5-qator
    cin >> x >> y;           //6-qator
    w=x+y;                   //7-qator
    cout << "W = " << w     //8-qator
    return 0;                //9-qator
}                             //10-qator
```

Error List					
5 Errors   0 Warnings   0 Messages					
Search Error List					
	Description	File	Line	Column	Project
1	error C2065: 'w' : undeclared identifier	Source.cpp	7	1	Project5
2	error C2065: 'w' : undeclared identifier	Source.cpp	9	1	Project5
3	error C2143: syntax error : missing ';' before 'return'	Source.cpp	9	1	Project5
4	IntelliSense: identifier "w" is undefined	Source.cpp	7	2	Project5
5	IntelliSense: expected a ','	Source.cpp	9	2	Project5

Дастурда 5та хато мавжуд, уларнинг қаторлари хам кўрсатилган. Биринчи хато w идентификатори мавжуд эмас, ушбу хатони тўғирлаш учун 5-қатор матнини қуйидагича ўзгартирилади:

```
int x, y, w; //5-qator
```

Сўнгра компиляция жараёнини ишга тушурилганда компилятор қуйидаги хатоларни кўрсатиб беради:



	Description	File	Line	Column	Project
1	error C2143: syntax error : missing ';' before 'return'	Source.cpp	9	1	Project5
2	IntelliSense: expected a ;'	Source.cpp	9	2	Project5

Кўриниб турибдики, биринчи хатонинг ўзи учта компиляция хатолигини юзага келтириб чиқарган.

Пробелларни ишлатишда ҳам алоҳида эътиборли бўлиш керак. Берилганлар киритилаётганда бир ёки бир нечта пробеллар берилганларни ажратиш учун ишлатилади. Шунингдек, калит сўзлар ва идентификаторларни ҳам бир-биридан ажратишда кўлланилади. Фақат, калит сўз ёки идентификаторларнинг ўзини ёзишда пробеллар умуман ишлатилмайди.

```
int a, b, c; //1-qator
double d, e, f; //2-qator
double birinchi; //3-qator
float ikkinchi; //4-qator
```

Ушбу намуналарда, 1-қатор тўғри ёзилган, “int” калит сўзи ва “a” идентификатори битта пробел ёрдамида ажратилган. Қолган идентификаторлардан олдин ҳам биттадан пробел кўйилган. 2-қаторда эса пробел нотўғри ишлатилган. “double” калит сўзини ёзишда нотўғри пробел кўйилган, натижада калит сўз ажралиб қолган ва бу компиляция жараёни хатолигига олиб келади. 3-қаторда “double” калит сўзи ва “birinchi” идентификатори орасида тўртта пробел мавжуд ва бу хатоликка олиб келмайди. 4-қаторда эса, “ikkinchi” идентификаторни ёзишда пробел нотўғри ишлатилган, аслида “ikkinchi” кўринишида ажратилмасдан ёзилиши керак эди. Ушбу қаторда ҳам компиляция хатолиги юзага келади.

Барча С++ инструкциялари нуқтали вергул – “;” билан тугалланиши зарур. Фигуралар кавслар (“{” ва “}”), хаттоки улар кўпинча бир қаторда ва ҳеч қандай дастур матнисиз келса ҳам инструкция эмас. Фигуралар кавслар программа қисмини бир бутунлик деб тушуниш учун ишлатилади. Вергул (“;”) одатда рўйхат элементларини ажратиш учун ишлатилади. Масалан, ўзгарувчилар эълон қилинаётганда бир турдаги бир нечта ўзгарувчини эълон қилишда вергулдан фойдаланилади.

Намуна учун қуйидаги C++ тилида ёзилган дастур қисмини кўрайлик:

```
int x, a;      // 1-qator
int y          // 2-qator
double z;     // 3-qator
y = w + x;   // 4-qator
```

Ушбу қаторлар компиляция қилинганда 2-қаторда компиляция хатолиги юзага келади. Чунки, 2-қаторда у ўзгарувчиси эълон қилинишидан кейин нуқтали вергул (“ ; ”) белгиси қўйилмаган. Иккинчи компиляция хатолиги 4-қаторда юзага келади. Бу қаторда w идентификатори ишлатилмоқда, аммо у эълон қилинмаган.

## 2.2. Тил семантикаси

Тилнинг маъносини берувчи қоидалар тўплами семантика дейилади. Масалан, арифметик операторлар бажарилиш кетма-кетлиги қоидаси семантик қоида.

Агар дастурда синтактик хатолар бўлса, компилятор бу ҳақида хабар беради. Лекин семантик хато бўлганда дастур ишлайди, лекин кутилган натижага эришиб бўлмайди. Масалан, қуйидаги икки қатор синтактик тўғри ёзилган, лекин маънолари турлича (турлича қиймат ҳосил бўлади):

```
2 + 3 * 5      // 1-qator
```

ва

```
(2 + 3) * 5    // 2-qator
```

Биринчи қаторда арифметик операторлар бажарилиш кетма-кетлиги қоида сига кўра кўпайтириш амали бажарилади, сўнгра қўшиш амали ишга тушади. Иккинчи қаторда эса, аввал қавс ичи бажарилади, сонлар қўшилади, кейин кўпайтириш амали ишлайди.

Қуйида икки хил кўринишдаги инструкцияларни кўрайлик:

```
const double A = 0.1; //o'zgartirish konstantasi
double x;           //santimetr uchun o'zgaruvchi
double y;           //millimetr uchun o'zgaruvchi
x = y * a;
```

ва

```
const double SANTIMETR_UCHUN_MILLIMETR = 0.1;
double santimetr;
double millimetr;
```



```
santimetr = millimetr * SANTIMETR_UCHUN_MILLIMETR;
```

### 2.3. Хужжатлашган идентификаторлар

Иккинчи кўринишдаги дастур инструкцияларидан фойдаланилганда `SANTIMETR_UCHUN_MILLIMETR` кўринишидаги идентификаторлар одатда хужжатлашган идентификаторлар деб аталади. Сабаби, дастурчи идентификатордан нима мақсадда фойдаланишни доим билиб туради. Биринчи кўринишдаги дастур инструкцияларида эса, идентификаторлардан нима мақсадда фойдаланилаётганлиги идентификаторларни эълон қилганда изоҳлар орқали бериб ўтилган. Дастур матнининг қолган қисмларида идентификаторнинг нима мақсадда ишлатилишини билиш учун доим дастур матни бошига ёки идентификатор эълон қилинган қисмига ўтиб изоҳ орқали ёзилган хужжатлаш қисмидан ўқиб тушуниш керак бўлади. Хужжатлашган идентификаторлар изоҳлардан фойдаланишни камайтириш учун ишлатилади. `olmasoni` – хужжатлашган идентификаторини тахлил қилайлик. Ушбу идентификаторни биргаликда қўлланилувчи сўз (run-together word) дейилади. Хужжатлашган идентификаторлардан фойдаланаётганда биргаликда қўлланилувчи сўзларни нотўғри қўллаш орқали хужжатлашда аниқликни камайтириш мумкин. Биргаликда қўлланилувчи сўзларни тушунарлироқ қўллаш учун бир қанча кўринишлар таклиф қилинган. Ҳар бир маънога эга сўзни бош ҳарф орқали ёзиш ёки улардан олдин таг чизик (“\_”) белгисини қўйиш мумкин. Масалан, тушунарлилиқ аниқлигини ошириш мақсадида `olmaSoni`, `olma_soni` кўринишидаги идентификаторлардан фойдаланиш мумкин. Константа ўзгарувчилардан фойдаланганда, уларни оддий ўзгарувчилардан фарқлаб туриш мақсадида барча ҳарфларни бош ҳарфлар ёрдамида ёзиш мақсадга мувофиқ.

### 2.4. Дастурни форматлаш

Яхши хужжатлашган дастур қисми сифатида фойдаланувчи дастур билан мулоқотда бўлганда (дастурни ишга тушурганда) унинг дастурни тушуниши ҳам назарда тутилади. Фойдаланувчи дастурни ишга тушурганда қандай берилганни киритиш кераклигини билмаса дастурдан фойдаланиш қийинлашади. Шунинг учун фойдаланувчига нима қилиш кераклигини ёки қандай тушунчага эга маълумот киритилиши ёки чиқарилаётганини кўрсатиб

бориш керак. Масалан, `int` туридаги `son` ўзгарувчиси учун куйидаги дастур қисми берилган бўлсин:

```
cout << "1 va 10 oralig'idagi butun sonni kiriting"
<< " va Enter tugmasini bosing" << endl;
cin >> son;
```

Ушбу дастур қисми ишга тушганда фойдаланувчи экранида куйидаги матн ҳосил бўлади:

```
1 va 10 oralig'idagi butun sonni kiriting va Enter tugmasini bosing
```

Ушбу қаторни кўрган фойдаланувчи 1 ва 10 оралиғидаги сонни киритиши, кейин Enter клавишини босиши кераклигини тушунади. Агар шу ёзувлар фойдаланувчига тақдим этилмаса, фойдаланувчи нимани киритишини билмасдан ўтираверади. Шунинг учун, одатда, фойдаланувчи киритиши лозим бўлган қийматдан олдин фойдаланувчига маълумот чиқариш орқали дастур тузилади.

Дастурчи ёзаётган дастур фақат унинг ўзи учун эмас, балки бошқалар учун ҳам тушунарли бўлиши керак. Шунинг учун дастурчи дастурини хужжатлаштириб тузиши керак. Яхши хужжатлашган дастурни биринчи ёзилганидан узоқ вақт ўтгандан кейин ҳам тушуниш ва ўзгартириш осон бўлади. Бунда хужжатлашган идентификаторлар ёки изоҳлардан фойдаланиш мумкин. Изоҳлар дастур қисмининг мақсади-мазмунини, дастурчи исмини (ким ёзганлигини) ва турли специфик маълумотларини уз ичига олиши керак. Шунингдек, дастур матнини ҳам турли қоидаларга асосланиб тузиш мақсадга мувофиқ. Масалан, ўзгарувчиларни эълон қилишда:

```
int gramm, tonna;

double x, y;

ва
int gramm,tonna;double x,y;
```

Иккала кўриниш ҳам хатосиз, тўғри ёзилган. Уларни тушунишда компьютер қийинчиликка дуч келмайди. Аммо дастурчи учун биринчи кўриниш тушунарлироқ. Сабаби, ўзгарувчиларнинг турлари алоҳида ажратилган, ўзгарувчилар ҳам пробел ёрдамида ажратиб ёзилган.

```
//Noto'g'ri formatlangan C++ dasturi.
```

```
#include <iostream>
#include <string>
```

```

using namespace std;
int main()
{
int son; double vazn;
string ism;
cout << "Son kiriting: "; cin >> son; cout << endl;
cout<<"son: "<<son<<endl;
cout<<"Ismingizni kiriting: "; cin>> ism;
cout<<endl; cout <<"Vazningizni kiriting: ";
cin>>vazn; cout<<endl;
cout<<"Ismingiz: "<<ism<<endl;cout<<"Vazningiz: "
<<vazn; cout <<endl;return 0;
}

```

**// To'g'ri formatlangan C++ dasturi.**

```

#include <iostream>
#include <string>
using namespace std;
int main()
{
int son;
double vazn;
string ism;
cout << "Son kiriting: ";
cin >> son;
cout << endl;
cout << "son: " << son << endl;
cout << "Ismingizni kiriting: ";
cin >> ism;
cout << endl;
cout << "Vazningizni kiriting: ";
cin >> vazn;
cout << endl;
cout << "Ismingiz: " << ism << endl;
cout << "Vazningiz: " << vazn << endl;

```

```
return 0;
}
```

Ушбу икки дастурдан кўриниб турибдики, нотўғри форматланган дастурни тушуниш тўғри форматланган дастурни тушунишга қараганда қийин.

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим мавзулари

1. Хатолар турлари.
2. Ўзгарувчилар ва ўзгармаслар.
3. Хатоларни бартараф этиш йўллари.
4. Дастурни форматлаш усуллари.

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
Синтактик қоидалар	Sintaktik rules	грамматик қоидаларга ўхшаш қоидалар тўплами
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
cout		экранга чиқариш оқими
семантика	semantics	тилнинг маъносини берувчи қоидалар тўплами
Компиляция	compilation	бажарилувчи файл ҳосил бўлиш жараёни

Лексема	lexeme	тилнинг ажралмайдиган қисмлари
Идентификатор	identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизик (‘_’) белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги

### Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур берилган километр узунликни метр ва сантиметрлар орқали ифодалайди. Дастур компиляция жараёнида синтактик тўғри ёзилган қаторда хатолик бор деб кўрсатмоқда. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тилида дастур коддини тўғирланг (индивидуал ҳолда).

### Назорат саволлари

1. Синтаксис – қандай тилнинг қоидалари?
2. C++ инструкциялари қандай белги билан тугалланиши зарур?
3. Тилнинг маъносини берувчи қоидалар тўплами қандай номланади?
4. SANTIMETR\_UCHUN\_MILLIMETR кўринишидаги идентификаторлар қандай аталади?
5. int туридаги son ўзгарувчиси учун дастур қисми кўрсатиб беринг.
6. Агар дастурда синтактик хатолар бўлса, компилятор бу ҳақида хабар берадими?
7. Арифметик операторлар бажарилиш кетма-кетлиги қоидаси қандай қоида?
8. Фигуралли қавслар нима учун ишлатилади?
9. Вергул (“,”) одатда нима учун ишлатилади?
10. “double” калит сўзи нима учун ишлатилади?

## Тест саволлари

1. C++ тилида қуйидаги амаллар бажарилганда жавоб неччи чиқади?

```
int c=25/3;
```

- a) 8
- b) 9
- c) 8.(3)
- d) 8.3

2. C++ тилидаги қуйидаги дастур қисмидаги синтактик хато кодни топинг?

```
int c=25/3; x = 5; z = 3; y = x - z; z = 2 * y + 3;
```

- a)  $x = 5;$
- b)  $y = x - z;$
- c)  $z = 2 * y + 3;$
- d) `int c=25/3;`

3.  $x=2$ ,  $y=5$  ва  $z=6$  га тенг бўлса, Қуйидаги амаллар кетма-кетлигини бажарилиши натижасида экранга нима чиқади?

```
cout << "x = " << x << ", y = " << y << ", z = " << z;
```

- a)  $x = 2, y = 5, z = 6$
- b)  $x = 22, y = 52, z = 62$
- c)  $x = <2, y = <5, z = <6$
- d)  $x = 2 y = 5 z = 6$

4.  $x=2$ ,  $y=5$  ва  $z=6$  га тенг бўлса, Қуйидаги амаллар кетма-кетлигини бажарилиши натижасида экранга нима чиқади?

```
cout << "x + y = " << x + y;
```

- a)  $x + y = 7$
- b)  $x + y = 11$
- c) 7
- d) 25

5.  $x=22$ ,  $y=5$  ва  $z=6$  га тенг бўлса, Қуйидаги амаллар кетма-кетлигини бажарилиши натижасида экранга нима чиқади?

```
cout << "x / z = " << x / y;
```

- a)  $x / z = 4$
- b)  $x / z = 3$
- c)  $x / z = 3.7$
- d)  $x / z = x / y$

6.  $x=13$ ,  $y=2$  ва  $z=11$  га тенг бўлса, Қуйидаги амаллар кетма-кетлигини бажарилиши натижасида экранга нима чиқади?

```
cout << (x + z) % y;
```

- a) 0
- b) 12
- c) 1
- d) 2

7.  $x=13$ ,  $y=2$  ва  $z=11$  га тенг бўлса, Қуйидаги амаллар кетма-кетлигини бажарилиши натижасида экранга нима чиқади?

```
cout << x + z % y;
```

- a) 14
- b) 0
- c) 1
- d) 2

8.  $x=13$ ,  $y=2$  ва  $z=11$  га тенг бўлса, Қуйидаги амаллар кетма-кетлигини бажарилиши натижасида экранга нима чиқади?

```
cout << (y + z) % y;
```

- a) 2
- b) 1
- c) 0
- d) 112

9.  $x=13$ ,  $y=2$  ва  $z=11$  га тенг бўлса, Қуйидаги амаллар кетма-кетлигини бажарилиши натижасида экранга нима чиқади?

```
cout << (x % y) % z ;
```

- a) 1
- b) 12
- c) 0
- d) 2

10. C++ тилидаги қуйидаги дастур қисмидаги синтактик хато кодни топинг?

```
int c=25/3, x = 5; z = 3; y = x - z; z = 2 * y + 3;
```

- a)  $z = 3;$
- b)  $y = x - z;$
- c)  $z = 2 * y + 3;$
- d) `int c=25/3;`



## Маъруза 3. Берилганлар турлари. С++ тилининг таянч турлари

### Режа:

1. Берилганларнинг оддий турлари
2. Белги тури
3. Мантикий тур
4. Ҳақиқий сон тури

Таянч иборалар: *include, cout, cin, Компиляция, Берилганлар, Идентификатор, int, double, char, байт.*

### 3.1. Берилганларнинг оддий турлари

С++ тилининг мақсади берилганлар устида амаллар бажариш орқали уларни бошқариш ҳисобланади. Турли дастурлар турли кўринишдаги берилганлар билан ишлайди. Масалан, ойлик иш ҳақини ҳисоблаш дастурида берилганлар кўшиш, айириш, кўпайтириш, бўлиш амаллари орқали қайта ишланади, берилганлар ишлаган соат ҳажми, иш ставкаси кўринишида аниқланади. Шу каби, талабаларнинг фамилияларини алфавит бўйича тартиблаш дастури фамилиялар кўринишидаги берилганлар билан ишлайди. Бу икки дастур икки хил кўринишдаги берилганлар билан ишлайди, биринчиси сонлар устида арифметик амаллар бажарса, иккинчиси сатр кўринишидаги берилганлар устида амаллар бажаради. Табиийки, бу иккала масалани бир хил амаллар орқали ечиб бўлмайди. Шунингдек, фамилия кўринишидаги берилганлар устида кўпайтириш ёки айириш амалини бажаришнинг имкони мавжуд эмас. Шуларни ҳисобга олган ҳолда, С++ тили берилганларни турларга ажратади. Ҳар бир берилганлар турлари устида фақатгина шу тур учун олдиндан аниқланган амалларни бажариш мумкин.

С++ тилида берилганлар учта категорияга ажратилади:

1. Берилганларнинг оддий турлари (таянч турлари).
2. Берилганларнинг структуралашган турлари.
3. Кўрсаткичлар.

Берилганларнинг оддий турлари С++ тилининг таянч турлари ҳисобланади, сабаби бу турлар берилганларнинг структуралашган турларини

кўришда асос бўлиб хизмат қилади. Берилганларнинг оддий турлари учта категорияга бўлинади:

1. Бутун сон турлари – бутун сонлар билан ишловчи турлар ёки ҳақиқий қисми бўлмаган сонлар турлари.
2. Сузувчи нуқтали турлар – ҳақиқий турдаги сонлар кўринишидаги берилганлар билан ишловчи турлар.
3. Санаб ўтилувчи тур – фойдаланувчи аниқлаган берилганлар тури.

Бутун сон турлари қуйидаги тўққизта берилганлар турларидан ташкил топган: `char`, `short`, `int`, `long`, `bool`, `unsigned char`, `unsigned short`, `unsigned int` ва `unsigned long`.

Нима сабабдан берилганларнинг оддий турларининг кўринишлари кўп деган савол пайдо бўлиши табиий. Ҳар бир тур ўзига хос параметрларга эга. Масалан, `char` берилганлар тури -128 ва 127 оралиғидаги сонларни ифодалаш учун ишлатилади, шунингдек бу тур ёрдамида белгаларни экранга чиқаришда фойдаланиш мумкин. `short` берилганлар тури -32768 ва 32767 оралиғидаги сонларни ифодалаш учун ишлатилади.

Бутун сон турлари, уларнинг байтлардаги ўлчамлари ва қийматларининг чегаралари қуйидаги жадвалда келтирилган.

Тур номи	Байтлардаги ўлчам	Қиймат чегараси
<code>bool</code>	1	true yoki false
<code>unsigned short int</code>	2	0..65535
<code>short int</code>	2	-32768..32767
<code>unsigned long int</code>	4	0..42949667295
<code>long int</code>	4	-2147483648..2147483647
<code>int (16 razryadli)</code>	2	-32768..32767
<code>int (32 razryadli)</code>	4	-2147483648..2147483647
<code>unsigned int (16 razryadli)</code>	2	0..65535
<code>unsigned int (32 razryadli)</code>	4	0..42949667295
<code>unsigned char</code>	1	0..255
<code>char</code>	1	-128..127

Бутун сон қийматларни қабул қиладиган ўзгарувчилар асосан `int` (бутун), `short` (қиска) ва `long` (узун) калит сўзлар билан аниқланади.

Ўзгарувчи қийматлари ишорали бўлиши ёки unsigned калит сўзи билан ишорасиз сон сифатида қаралиши мумкин. С++ тилида бутун сонлар худди математикадаги каби кўринишларда бўлади:

-6728, -13, 0, 47, +485.

Бутун сонлар қуйидаги форматларда бўлади:

- ўнлик сон;
- саккизлик сон;
- ўн олтилик сон.

Ўнлик ўзгармас 0 рақамдан фарқли рақамдан бошланувчи рақамлар кетма-кетлиги ва 0 ҳисобланади: 0; 123; 7987; 11.

Манфий ўзгармас - бу ишорасиз ўзгармас бўлиб, унга фақат ишорани ўзгартириш амали қўлланилган деб ҳисобланади.

Саккизлик ўзгармас 0 рақамдан бошланувчи саккизлик санок системаси (0, 1, ... , 7) рақамларидан ташкил топган рақамлар кетма-кетлиги: 023; 0777; 0.

Ўн олтилик ўзгармас 0x ёки 0X белгиларидан бошланадиган ўн олтилик санок системаси рақамларидан иборат кетма-кетлик ҳисобланади:

0x1A; 0X9F2D; 0x23.

Ҳарф белгилар ихтиёрий регистрларда берилиши мумкин.

Компилятор соннинг қийматига қараб унга мос турни белгилайди. Агар тилда белгиланган турлар дастур тузувчини қаноатлантормаса, у ошкор равишда турни кўрсатиши мумкин. Бунинг учун бутун ўзгармас рақамлари охирига, пробелсиз л ёки L (лонг), у ёки U (унсигнед) ёзилади. Зарур ҳолларда битта ўзгармас учун бу белгиларнинг иккитасини ҳам ишлатиш мумкин:

451u, 012U1, 0xA2L.

Бутун сонлар билан ишлаганда агар сон мусбат бўлса, “+” белгисини қўйиш шарт эмас. Шунингдек, нуқта ва вергул бутун сонлар учун ажратувчи бўлиб хизмат қилади. Компилятор 31.258 ёки 31,258 кўринишидаги ёзувни 31 ва 258 дан иборат иккита бутун сон деб тушунади.

### 3.2. Белги тури

Белги туридаги ўзгарувчилар char калит сўзи билан берилади ва улар ўзида белгининг ASCII кодини сақлайди. Яъни, компилятор белгининг кодини сақлайди, чоп этаётганда эса белги кўринишида экранга чиқарилади. Белги туридаги қийматлар нисбатан мураккаб бўлган тузилмалар - сатрлар,

белгилар массивлари ва ҳақозаларни ҳосил қилишда ишлатилади. Белги ўзгармаслар кўштирноқ (‘,’-апострофлар) ичига олинган алоҳида белгилардан ташкил топади. Белги ўзгармас учун хотирада бир байт жой ажратилади ва унда бутун сон кўринишидаги белгининг ASCII коди жойлашади. Қуйидагилар белги ўзгармасларга мисол бўлади: ‘e’, ‘@’, ‘7’, ‘z’, ‘w’, ‘+’, ‘\*’, ‘a’, ‘\$’. Апострофлар орасида фақат битта белги жойлашиши керак. ‘abc’ белги ҳисобланмайди. С++ дастурда бундай кўринишда ишлатилганда компилятор хатолик ҳақида хабар беради.

Айрим белги ўзгармаслар ‘\’ белгисидан бошланади, бу белги биринчидан, график кўринишга эга бўлмаган ўзгармасларни белгилайди, иккинчидан, махсус вазифалар юкланган белгилар - апостроф белгиси(\), савол белгисини (“?”), тескари ён чизиқ белгисини (“\”) ва иккита кўштирноқ белгисини (“”) чоп қилиш учун ишлатилади. Ундан ташқари, бу белги орқали белгини кўринишини эмас, балки ошкор равишда унинг ASCII кодини саккизлик ёки ўн олтилик шаклда ёзиш мумкин. Бундай белгидан бошланган белгилар escape кетма-кетликлар дейилади.

С++ тилида escape-белгилар жадвали:

escape-белгилар	Ички код (16 лик сон)	Номи	Амал
\\	0x5S	\	Тескари ён чизиқни чоп этиш
\'	0x27	'	Апострофни чоп этиш
\"	0x22	"	Кўштирноқни чоп этиш
\?	0x3F	?	Сўроқ белгиси
\a	0x07	bel	Товуш сигналинини бериш
\b	0x08	bs	Курсорни 1 белги орқага қайтариш
\f	0x0C	ff	Саҳифани ўтказиш
\n	0x0A	lf	Қаторни ўтказиш
\r	0x0D	cr	Курсорни айна қаторнинг бошига қайтариш
\t	0x09	ht	Навбатдаги табуляция жойига ўтиш
\v	0x0D	vt	Вертикал табуляция
\000	000		Белги саккизлик коди билан берилганда
\xNN	0xNN		Белги ўн олтилик коди билан берилганда

Мисол:

```
#include <iostream>
using namespace std;
int main()
{
char belgi=99;
cout<< "Belgi:\t" << belgi << '\n';
cout<<"Dastur tugadi!";
return 0;
}
```

Ушбу дастур ишга тушганда экранга қуйидаги натижа чиқади:

```
Belgi:  c
Dastur tugadi!
```

### 3.3. Мантиқий тур

Бу турдаги ўзгарувчи `bool` калит сўзи билан эълон қилинади. У турдаги ўзгарувчи 1 байт жой эгаллайди ва 0 (`false`, ёлғон) ёки 0 қийматидан фарқли қиймат (`true`, рост) қабул қилади. Мантиқий турдаги ўзгарувчилар қийматлар ўртасидаги муносабатларни ифодаладиган мулоҳазаларни рост ёки ёлғон эканлигини тавсифлашда қўлланилади ва улар қабул қиладиган қийматлар математик мантиқ қонуниятларига асосланади.

*Математик мантиқ* – фикрлашнинг шакли ва қонуниятлари ҳақидаги фан. Унинг асосини мулоҳазалар ҳисоби ташкил қилади. *Мулоҳаза* – бу ихтиёрий жумла бўлиб, унга нисбатан рост ёки ёлғон фикрни билдириш мумкин. Масалан “ $3 > 2$ ”, “5 - жуфт сон”, “Лондон-Италия пойтахти” ва ҳоказо. Лекин “0.000001 - кичик сон” жумласи мулоҳаза ҳисобланмайди, чунки “кичик сон” тушунчаси жуда ҳам нисбий, яъни кичик сон деганда қандай сонни тушуниш кераклиги аниқ эмас. Шунинг учун юқоридаги жумлани рост эки ёлғонлиги ҳақида фикр билдириш қийин.

Мулоҳазаларнинг ростлиги ҳолатларга боғлиқ равишда ўзгариши мумкин. Масалан “*бугун – чоршанба*” жумласини рост ёки ёлғонлиги айти каралаётган кунга боғлиқ. Худди шундай “ $x < 0$ ” жумласи  $x$  ўзгарувчисининг айти пайтдаги қийматига мос равишда рост ёки ёлғон бўлади.

C++ тилида мантикий тур номи англиялик математик Жорж Бул шарафига `bool` сўзи билан ифодаланган. Мантикий амаллар “*Бул алгебраси*” дейилади.

### 3.4. Ҳақиқий сон тури

Ҳақиқий сонлар билан ишлаш учун C++ тилида сузувчи нуқтали берилганлар турлари мавжуд. Ҳақиқий ўзгармаслар – сузувчи нуқтали сон бўлиб, у икки хил форматда берилиши мумкин:

- ўнлик фиксирланган нуқтали форматда. Бу кўринишда сон нуқта орқали ажратилган бутун ва каср қисмлар кўринишида бўлади. Соннинг бутун ёки каср қисми бўлмаслиги мумкин, лекин нуқта албатта бўлиши керак. Фиксирланган нуқтали ўзгармасларга мисоллар: 24.56; 13.0; 66.; .87;
- экспоненциал шаклда ҳақиқий ўзгармас 6 қисмдан иборат бўлади:
  1. бутун қисми (ўнли бутун сон);
  2. ўнли каср нуқта белгиси;
  3. каср қисми (ўнлик ишорасиз ўзгармас);
  4. экспонента белгиси ‘e’ ёки ‘E’;
  5. ўн даражаси кўрсаткичи (ўнли бутун сон);
  6. қўшимча белгиси (‘F’ ёки ‘f’, ‘L’ ёки ‘l’).

Экспоненциал шаклдаги ўзгармас сонларга мисоллар:  $1e2$ ;  $5e+3$ ;  $.25e4$ ;  $31.4e-1$ .

Ҳақиқий сонлар ва уларнинг экспоненциал кўринишлари билан қуйидаги жадвал орқали танишиш мумкин:

Ҳақиқий сон	Экспоненциал кўриниш
75.924	7.592400E1
0.18	1.800000E-1
0.0000453	4.530000E-5
-1.482	-1.482000E0
7800.0	7.800000E3

Ҳақиқий сонлар `float` ёки `double` калит сўзи билан эълон қилинади. Бу турдаги ўзгарувчи учун хотирада 4 байт жой ажратилади ва <ишора><тартиб><мантисса> қолипда сонни сақлайди. Агар касрли сон жуда катта (кичик) қийматларни қабул қиладиган бўлса, у хотирада 8 ёки 10

байтда иккиланган аниқлик кўринишида сақланади ва мос равишда double ва long double калит сўзлари билан эълон қилинади. Охирги ҳолат 32-разрядли платформалар учун ўринли.

Тур номи	Байтлардаги ўлчами	Қиймат чегараси
float	4	-3.4E+38..3.4E+38
double	8	-1.7E+308..1.7E+308
long double (32 razryadli)	10	-3.4e-4932..3.4e4932

```
#include <iostream>
using namespace std;
int main()
{
const double pi=3.1415;
const int Radius=3;
double Yuza=0;
Yuza=pi*Radius*Radius;
cout<<Yuza<<'\n';
cout<<"Dastur tugadi!";
return 0;
}
```

#### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

#### Мустақил таълим мавзулари

1. Бутун сон тур кўринишлари.
2. Ҳақиқий сон тур кўринишлари.
3. Мантиқий турдан фойдаланиш.
4. Тур ўзгарувчилари устида амаллар.

## Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
cout		экранга чиқариш оқими
cin		экрандан киритиш оқими
Компиляция	compilation	бажарилувчи файл хосил бўлиш жараёни
Берилганлар	variable	дастур ишлаши учун керакли қийматлар
Идентификатор	identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизиқ ('_') белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги.
int		бутун сон кўринишидаги берилганларнинг тури
double		хақиқий сон кўринишидаги берилганларнинг тури
char		белги кўринишидаги берилганларнинг тури
байт		компьютер хотираси ўлчов бирлиги



## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур бўлинманинг каср қисмини топиши керак. Лекин дастур нотўғри ишламоқда, соннинг бўлинмаси доим бутун сонга айланмоқда. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили имкониятларидан келиб чиқиб дастур коддини тақдим этинг (индивидуал ҳолда).

## Назорат саволлари

1. Ҳақиқий сон тури нима?
2. Берилганларнинг структуралашган турлари қандай?
3. Бутун сон турлари нима?
4. Сузувчи нуқтали турлар нима?
5. Санаб ўтилувчи тур нима?
6. Ҳақиқий сонлар қандай калит сўзи билан эълон қилинади?
7. Ўнлик фиксирланган нуқтали формат деганда нимани тушунилади?
8. Экспоненциал шаклда ҳақиқий ўзгармас неча қисмдан иборат бўлади ва уларга мисол кўрсатинг.
9. Ҳарф белгилар қандай регистрларда берилиши мумкин?
10. Компилятор нимага қараб унга мос турни белгилайди?

## Тест саволлари

1. short туридаги ўзгарувчининг қабул қиладиган қийматлар оралиғи?
  - a) -32768...32767
  - b) 0..65535
  - c) 0..32767
  - d) -2147483648.. 2147483647

2. C++ тилида char туридаги ўзгарувчиларнинг қабул қиладиган қийматлар оралиғи тўғри кўрсатилган жавобни аниқланг.
- a) 0..255
  - b) -128..127
  - c) 0..32
  - d) -32768..32767
3. C++ тилида double туридаги ўзгарувчилар хотирада қанча жой эгаллайди?
- a) 8байт
  - b) 4 байт
  - c) 2 байт
  - d) 1 байт
4. Идентификатор сифатида фойдаланиш мумкин бўлмаган жавобни кўрсатинг.
- a) While
  - b) katta
  - c) kichik
  - d) B12
5. Идентификатор сифатида фойдаланиш мумкин бўлган жавобни кўрсатинг.
- a) dasTuR
  - b) int
  - c) double
  - d) char
6. C++да операторлар қандай белги билан ажратилади?
- a) ;
  - b) “ ” (пробел)
  - c) :
  - d) ,

7. 32 разрядли тизимда int тури компьютер хотирасидан қанча жой эгаллайди.
- a) 4 байт
  - b) 2 байт
  - c) 6 байт
  - d) 8 байт
8. Қайси тоифа форматдаги қиймат компьютер хотирасидан энг катта жой эгаллайди.
- a) long double
  - b) double
  - c) int
  - d) float
9. C++ тилида бутун сон туридаги ўзгарувчилар қайси калит сўз ёрдамида аниқланади?
- a) int
  - b) float
  - c) void
  - d) double
10. C++ тилида белги туридаги ўзгарувчилар қайси калит сўз ёрдамида аниқланади?
- a) char
  - b) string
  - c) double
  - d) void

## Маъруза 4. Ўзгарувчилар ва ифодалар

### Режа:

1. Таянч арифметик амаллар
2. Турни бошқа турга келтириш
3. Константа ўзгармаслар
4. Ифодаларда ўзгарувчилардан фойдаланиш

Таянч иборалар: *include, cout, cin, Компиляция, Унар амал, Бинар амал, Идентификатор, Ўзгарувчи, Константа.*

### 4.1. Таянч арифметик амаллар

Берилганларни қайта ишлаш учун C++ тилида амалларнинг жуда кенг мажмуаси аниқланган. Амал – бу қандайдир ҳаракат бўлиб, у битта (унар) ёки иккита (бинар) операндлар устида бажарилади, ҳисоб натижаси унинг қайтарувчи қиймати ҳисобланади.

Таянч арифметик амалларга қўшиш (+), айириш (–), кўпайтириш (\*), бўлиш (/) ва бўлишнинг қолдиғини олиш (%) амалларини келтириш мумкин. Қўшиш, айириш, кўпайтириш, бўлиш амалларини бутун ва ҳақиқий турдаги сонли берилганлар билан ишлатиш мумкин. Бўлишнинг қолдиғини олиш амалини эса фақат бутун турдаги сонли берилганлар билан ишлатилади. Шунингдек, бўлиш амалини бутун сонли берилганлар устида амалга оширилса, натижа сифатида бўлишнинг бутун қисми қайтарилади.

a.  $-5$

b.  $8 - 7$

c.  $3 + 4$

d.  $2 + y * 5$

e.  $5.6 + 6.2 * 3$

f.  $x + 2 * 5 + 6 / y$

Намуналарда “x” ва “y” номаълум сонлар. Арифметик ифода арифметик операторлар (амаллар) ва сонлардан тузилади. Ифодада қатнашган сонлар (номаълум сонлар ҳам) операнд деб аталади. “a” намунада “–” амали 5 сонининг манфийлигини аниқлаш учун ишлатилмоқда. Бу ифодада битта операнд мавжуд. Фақатгина битта операнди бор бўлган арифметик оператор

(амал) унар оператор деб аталади. “b” намунада “-” амали саккиз сонидан етти сонини айириш учун ишлатилмоқда. Бу арифметик ифодада “-” амалининг иккита операнди бор – 8 ва 7. Иккита операнди мавжуд операторлар бинар операторлар дейилади.

```
#include <iostream>
using namespace std;
int main()
{
cout << "2 + 5 = " << 2 + 5 << endl;
cout << "13 + 89 = " << 13 + 89 << endl;
cout << "34 - 20 = " << 34 - 20 << endl;
cout << "45 - 90 = " << 45 - 90 << endl;
cout << "2 * 7 = " << 2 * 7 << endl;
cout << "5 / 2 = " << 5 / 2 << endl;
cout << "14 / 7 = " << 14 / 7 << endl;
cout << "34 % 5 = " << 34 % 5 << endl;
cout << "4 % 6 = " << 4 % 6 << endl;
cout << "5.0 + 3.5 = " << 5.0 + 3.5 << endl;
cout << "3.0 + 9.4 = " << 3.0 + 9.4 << endl;
cout << "16.3 - 5.2 = " << 16.3 - 5.2 << endl;
cout << "4.2 * 2.5 = " << 4.2 * 2.5 << endl;
cout << "5.0 / 2.0 = " << 5.0 / 2.0 << endl;
cout << "34.5 / 6.0 = " << 34.5 / 6.0 << endl;
cout << "34.5 / 6.5 = " << 34.5 / 6.5 << endl;
return 0;
}
```

**Натижа:**

```
2 + 5 = 7
13 + 89 = 102
34 - 20 = 14
45 - 90 = -45
2 * 7 = 14
5 / 2 = 2
14 / 7 = 2
34 % 5 = 4
```

$$4 \% 6 = 4$$

$$5.0 + 3.5 = 8.5$$

$$3.0 + 9.4 = 12.4$$

$$16.3 - 5.2 = 11.1$$

$$4.2 * 2.5 = 10.5$$

$$5.0 / 2.0 = 2.5$$

$$34.5 / 6.0 = 5.75$$

$$34.5 / 6.5 = 5.30769$$

5/2 арифметик ифодада иккала операнд ҳам бутун сон бўлгани учун бўлиш амали бўлиш натижасининг бутун қисмини олиш учун ишламоқда. Агар натижа сифатида ҳақиқий сонни қайтариш керак бўлса, операндларнинг камида биттаси ҳақиқий сон турида бўлиши керак,  $5.0/2.0$ ,  $5./2$ ,  $5/2$ . кўринишларида ишлатиш мумкин. Агар соннинг каср қисми бўлмаса, лекин уни ҳақиқий сон сифатида ишлатиш керак бўлса, 5.0 ёки 5. Кўринишида ёзиш мумкин. Шу сабабли ҳам натижа икки чиққан.  $34\%5$  ифодасида эса, 34 сонини 5 сонига бўлганда 6 бутун сон ва 4 қолдиқ сон чиқади, % амали қолдиқ қисмини қайтаришини инобатга олган ҳолда натижа сифатида 4 сони экранга чиқарилган. Бўлишнинг қолдиғини олиш амалини манфий сонлар билан ишлатаётганда жуда эҳтиёткор бўлиш керак.

Арифметик ифодада бир нечта операторлар (амаллар) қатнашганда компилятор амаллар кетма-кетлигини уларнинг устунлигига қараб бажаради. Аввал кўпайтириш, бўлиш, бўлишнинг қолдиғини олиш амаллариЮ сўнгра қўшиш ва айириш амаллари бажарилади. Масалан:  $3 * 7 - 6 + 2 * 5 / 4 + 6$  ифодаси учун амалларнинг бажарилиш кетма-кетлигини қуйидагича тушуниш мумкин:

$$3 * 7 - 6 + 2 * 5 / 4 + 6$$

$$= (((3 * 7) - 6) + ((2 * 5) / 4)) + 6 \quad (* \text{ бажарилади})$$

$$= ((21 - 6) + (10 / 4)) + 6 \quad (/ \text{ бажарилади. Операндлар бутун сонлигини ҳисобга олинган ҳолда})$$

$$= ((21 - 6) + 2) + 6 \quad (- \text{ бажарилади})$$

$$= (15 + 2) + 6 \quad (\text{биринчи} + \text{бажарилади})$$

$$= 17 + 6 \quad (+ \text{ бажарилади})$$

$$= 23$$

Агар ифодадаги барча операндлар бутун сонлардан ташкил топган бўлса, бу ифода бутун сонли ифода дейилади. Агар ифода операндлари ҳақиқий сонлардан ташкил топган бўлса, ҳақиқий сонли ёки сузувчи нуқтали сонли ифода дейилади.

```
2 + 3 * 5
3 + a - b / 7
a + 2 * (b - c) + 18
12.8 * 17.5 - 34.50
x * 10.5 + y - 16.2
```

Агар ифодада ҳам бутун турдаги, ҳам ҳақиқий турдаги сонлар ёки ўзгарувчилар қатнашса бундай ифода аралаш ифода дейилади.

```
2 + 3.5
6 / 4 + 3.9 - a / b
5.4 * 2 - 13.6 + 18 / 2
```

Аралаш ифодада ҳар бир операторлар (амаллар) алоҳида қисм ифода сифатида қараб ҳисобланади. Агар қисм ифода операндлари бир хил турда бўлса, натижа ҳам шу турда бўлади, агар операндлари турлари фарқли бўлса, натижа ҳақиқий сон кўринишида бўлади.

```
#include <iostream>
using namespace std;
int main()
{
cout << "3 / 2 + 5.5 = " << 3 / 2 + 5.5 << endl;
cout << "15.6 / 2 + 5 = " << 15.6 / 2 + 5 << endl;
cout << "4 + 5 / 2.0 = " << 4 + 5 / 2.0 << endl;
cout << "4 * 3 + 7 / 5 - 25.5 = "
<< 4 * 3 + 7 / 5 - 25.5
<< endl;
return 0;
}
```

**Натижа:**

```
3 / 2 + 5.5 = 6.5
15.6 / 2 + 5 = 12.8
4 + 5 / 2.0 = 6.5
```

$$4 * 3 + 7 / 5 - 25.5 = -12.5$$

Аралаш ифодада ифоданинг натижаси компилятор томонидан ҳисобланади. Бутун сонли қийматлар каср қисми нолга тенг бўлган ҳақиқий сонга айлантирилади. Бу жараён бир турни бошқа турга келтириш дейилади. C++ тилида бир турни бошқа турга келтиришнинг ошкор ва ошкормас йўллари мавжуд.

## 4.2. Турни бошқа турга келтириш

Умуман олганда, турни бошқа турга ошкормас келтириш ифодада ҳар хил турдаги ўзгарувчилар қатнашган ҳолларда амал қилади (аралаш турлар арифметикаси). Айрим ҳолларда, хусусан таянч турлар билан боғлиқ турга келтириш амалларида хатоликлар юзага келиши мумкин. Масалан, ҳисоблаш натижасидаги соннинг хотирадан вақтинча эгаллаган жойи узунлиги, уни ўзлаштирадиган ўзгарувчи учун ажратилган жой узунлигидан катта бўлса, қийматга эга разрядларни йўқотиш ҳолати юз беради.

Ошкор равишда турга келтиришда, ўзгарувчи олдида қавс ичида бошқа тур номи ёзилади:

```
#include <iostream>
using namespace std;
int main()
{
    int Integer_1=54;
    int Integer_2;
    float Floating=15.854;
    Integer_1=(int)Floating;           // ошкор келтириш
    Integer_2=Floating;               // ошкормас келтириш
    cout << "Yangi Integer (Oshkor): "<<Integer_1<<"\n";
    cout << "Yangi Integer (Oshkormas): "
        <<Integer_2<<"\n";
    return 0;
}
```



Дастур натижаси қуйидаги кўринишида бўлади:

```
Yangi Integer(Oshkor) : 15
```

```
Yangi Integer(Oshkormas) : 15
```

Масала. Берилган белгининг ASCII коди чоп этилсин. Масала белги туридаги қийматни ошкор равишда бутун сон турига келтириб чоп қилиш орқали ечилади.

Дастур матни:

```
#include <iostream.h>
int main()
{
    unsigned char A;
    cout << "Belgini kiriting: ";
    cin >> A;
    cout << '\'' << A << "'-belgi ASCII kodi="
        << (int)A << '\n';
    return 0;
}
```

Дастурнинг

```
Belgini kiriting:
```

сўровига

```
A <enter>
```

амали бажарилса, экранга

```
'A'-belgi ASCII kodi=65
```

сатри чоп этилади.

Шунингдек, cast оператори ёрдамида ҳам ошкор равишда бир турни бошқа турга келтириш мумкин:

```
static_cast<dataTypeName>(expression)
```

expression – қийматини бошқа турга ўтказиш лозим бўлган ифода.

dataTypeName – ифодани ўтказиш лозим бўлган тур номи.

```
#include <iostream>
```

```
using namespace std;
```

```

int main()
{
cout << "static_cast<int>(7.9) = "
    << static_cast<int>(7.9) << endl;
cout << "static_cast<int>(3.3) = "
    << static_cast<int>(3.3) << endl;
cout << "static_cast<double>(25) = "
    << static_cast<double>(25) << endl;
cout << "static_cast<double>(5 + 3) = "
    << static_cast<double>(5 + 3) << endl;
cout << "static_cast<double>(15) / 2 = "
    << static_cast<double>(15) / 2 << endl;
cout << "static_cast<double>(15 / 2) = "
    << static_cast<double>(15 / 2) << endl;
cout << "static_cast<int>(7.8 + static_cast<double>(15)
/ 2) = "
    << static_cast<int>(7.8 + static_cast<double>(15) /
2)
    << endl;
cout << "static_cast<int>(7.8 + static_cast<double>(15 /
2)) = "
    << static_cast<int>(7.8 + static_cast<double>(15 /
2))
    << endl;
return 0;
}

```

Дастур натижаси қуйидаги кўринишида бўлади:

```
static_cast<int>(7.9) = 7
static_cast<int>(3.3) = 3
static_cast<double>(25) = 25
static_cast<double>(5 + 3) = 8
static_cast<double>(15) / 2 = 7.5
static_cast<double>(15 / 2) = 7
static_cast<int>(7.8 + static_cast<double>(15) / 2)
= 15
static_cast<int>(7.8 + static_cast<double>(15 / 2))
= 14
```

Шунингдек, `cast` оператори ёрдамида белгилар бошқа турга келтириш мумкин. Бунда, белгининг ASCII жадвалидаги коди орқали бошқа турга келтирилади. Яъни:

```
static_cast<int>('A') => 65
static_cast<int>('8') => 56.
static_cast<char>(65) => 'A'
static_cast<char>(56) => '8'.
```

### 4.3. Константа ўзгармаслар

C++ тилида тузилган дастурнинг асосий мақсади ҳисоблаш ишларини амалга ошириш ва берилганларни бошқариш ҳисобланади. Берилганлар устида бирор иш бажарилишидан олдин улар асосий хотирадан жой олган бўлиши керак. Бунинг учун ўзгарувчилар ёки константалар ишлатилади.

Константа ўзгармаслар `const` калит сўзи орқали эълон қилинади:

```
const dataType identifier = value;
```

`dataType` – тур номи

`identifier` – идентификатор

`value` – қиймат

```
const double KATTALIK = 2.54;
```

```
const int TALABALAR_SONI = 20;
const char PROBEL = ' ';
```

Ўзгарувчи – дастур объекти бўлиб, хотирадаги бир нечта ячейкаларни эгаллайди ва берилганларни сақлаш учун хизмат қилади. Ўзгарувчи номга, ўлчамга ва бошқа атрибутларга - кўриниш соҳаси, амал қилиш вақти ва бошқа хусусиятларга эга бўлади. Ўзгарувчиларни ишлатиш учун улар албатта эълон қилиниши керак. Эълон натижасида ўзгарувчи учун хотирадан қандайдир соҳа захираланади, соҳа ўлчами эса ўзгарувчининг конкрет турига боғлиқ бўлади. Шунини қайд этиш зарурки, битта турга турли аппарат платформаларда турлича жой ажратилиши мумкин.

#### 4.4. Ифодаларда ўзгарувчилардан фойдаланиш

Ўзгарувчи эълони унинг турини аниқловчи калит сўзи билан бошланади ва '=' белгиси орқали бошланғич қиймат берилади (шарт эмас). Битта калит сўз билан бир нечта ўзгарувчиларни эълон қилиш мумкин. Бунинг учун ўзгарувчилар бир-биридан ',' белгиси билан ажратилади. Эълонлар ';' белгиси билан тугайди.

```
dataType identifier, identifier, . . . ;
double Yuza;
int soni;
char ch;
int x, y;
```

Ўзгарувчиларга ифода '=' белгиси орқали юкланади. Бунда ифоданинг натижавий қийматининг тури ўзгарувчи тури билан мос келиши керак. Акс ҳолда, агар компилятор турлар ўртасида бир турдан бошқа турга келтиришни амалга ошира олса, ифоданинг қиймат ўзгарувчининг турига ўтказиб юкланади, акс ҳолда компилятор хатолигига олиб келади.

```
#include <iostream>
using namespace std;
int main()
{
int num1, num2, num3;
```

```

double qiymat;
char belgi;
num1 = 4;
cout << "num1 = " << num1 << endl;
num2 = 4 * 5 - 11;
cout << "num2 = " << num2 << endl;
num3 = 4.5 * 5 - 9;
cout << "num3 = " << num3 << endl;
qiymat = 0.02 * 1000;
cout << "qiymat = " << qiymat << endl;
belgi = 'D';
cout << "belgi = " << belgi << endl;
return 0;
}

```

Дастур натижаси қуйидаги кўринишида бўлади:

```

num1 = 4
num2 = 9
num2 = 13
qiymat = 20
belgi = D

```

C++ тилида `num = num + 2;` кўринишидаги ифода `num` ўзгарувчисининг қийматини иккитага ошириш кераклигини билдиради. Ифода бажарилишидан олдин `num` ўзгарувчисига қиймат берилган бўлиши керак. Акс ҳолда дастур кутилмаган қиймат чоп этади. Сабаби, ўзгарувчи эълон қилинганда унга ҳеч қандай бошланғич қиймат берилмайди, шу сабабли компилятор учун `num` ўзгарувчисининг бошланғич қиймати мавжуд эмас.

```

int num1, num2, num3;           //1-qator
num1 = 18;                       //2-qator

```

```

num1 = num1 + 27;           //3-qator
num2 = num1;                //4-qator
num3 = num2 / 5;           //5-qator
num3 = num3 / 4;           //6-qator

```

Код қисмидаги ўзгарувчиларнинг компилятор учун қандай кетма-кетликда қиямат олишлари қуйидаги жадвалда келтирилган:

	<b>Ўзгарувчиларнинг қийматлари</b>	<b>Изоҳ</b>
1-қатор	num1=? num2=? num3=?	
2-қатор	num1=18 num2=? num3=?	
3-қатор	num1=45 num2=? num3=?	num1 + 27 = 18 + 27 = 45. Натижа num1 ўзгарувчисига, унинг олдинги қийматининг ўрнига ёзилади.
4-қатор	num1=45 num2=45 num3=?	num1 ўзгарувчисининг қиймати num2 ўзгарувчисига юкланади
5-қатор	num1=45 num2=45 num3=9	num2 / 5 = 45 / 5 = 9. Чиққан натижа num3 ўзгарувчисига юкланади
6-қатор	num1=45 num2=45 num3=2	num3 / 4 = 9 / 4 = 2. Натижа num3 ўзгарувчисига, унинг олдинги қийматининг ўрнига ёзилади.

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим мавзулари

1. Ифодаларнинг дастурлаш тилида ёзилиши.
2. Бир турдан бошқа турга ўтказиш.
3. Дастурда константа ўзгармасларнинг аҳамияти.
4. Арифметик амаллардай фойдаланиш.

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
cout		экранга чиқариш оқими
cin		экрандан киритиш оқими
Компиляция	compilation	бажарилувчи файл хосил бўлиш жараёни
Унар амал	Unar	битта операнд устида бажарилувчи амал
Бинар амал	Binary	иккита операнд устида бажарилувчи амал
Идентификатор	identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизик (‘_’)

		белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги
Ўзгарувчи	variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
Константа	const	дастур давомида қиймати ўзгармайдиган берилган

### Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур киритилган белгининг ASCII жадвалидаги кодини топиши керак. Лекин дастур белгининг ўзини чоп этмоқда. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

### Назорат саволлари

1. Аралаш ифодада қандай ҳисобланади?
2. Берилган белгининг ASCII коди чоп этилсин. Масала белги туридаги қийматни қандай ечилади?
3. Қайси оператор ёрдамида ошкор равишда бир турни бошқа турга келтириш мумкин?
4. cast оператори ёрдамида белгилар бошқа турга келтириш мумкинми?
5. C++ тилида тузилган дастурнинг асосий мақсади нима?
6. Ўзгарувчи нима?
7. Ўзгарувчиларга ифода қандай белги орқали юкланади?
8. C++ тилида `num = num + 2;` кўринишидаги ифода нимани билдиради?



9. Код қисмидаги ўзгарувчиларнинг компилятор учун қандай кетма-кетликда қиймат олишларини жадвалини ёзинг.

10. C++ тилида бир турни бошқа турга келтиришнинг қандай йўллари мавжуд?

### Тест саволлари

1. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `cout<< "3 / 2 + 5.5 = " << 3 / 2 + 5.5 <<endl;`

- a)  $3 / 2 + 5.5 = 6.5$
- b) 6.0
- c)  $2 + 5.5$
- d) 6.5

2. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `cout<<static_cast<int>(7.9);`

- a) 7
- b) 7.0
- c) 0.9
- d) 7.9

3. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `cout<<static_cast<int>(7.8 + static_cast<double>(15 / 2));`

- a) 14
- b) 15
- c) 14.5
- d) 15.3

4. Қуйидаги дастур кодининг натижасини топинг.

`num2 = 4 * 5 - 11; cout<< "num2 = " << num2 <<endl;`

- a) `num2 = 9`
- b)  $4 * 5 - 11 = 9$
- c) `num2 = -24`
- d) `num2 = 19`

5. Қуйидаги дастур кодининг натижасини топинг.

```
num2 = 4 / 5 - 11; cout<< num2 <<endl;
```

- a) -11
- b) 10.2
- c) -10.2
- d) 11

6. Қуйидаги дастур кодининг натижасини топинг.

```
num2 = 4 % 5 + 5%4; cout<< num2 <<endl;
```

- a) 5
- b) 0
- c) 9
- d) 2.05

7. Қуйидаги дастур кодининг натижасини топинг.

```
num2 = 4 / 5 - 4 % 5; cout<< num2 <<endl;
```

- a) -4
- b) 4
- c) 5
- d) 11

8. Қуйидаги дастур кодининг натижасини топинг.

```
num2 = 4 *3 + 7 / 5 - 25.5; cout<< num2 <<endl;
```

- a) -12.5
- b) 10.5
- c) -12
- d) 12

9. Қуйидаги дастур кодининг натижасини топинг.

```
num = 6; num = num + 2; cout<< num <<endl;
```

- a) 8
- b) 10
- c) -2
- d) 2

10. Қуйидаги ифодаларнинг ичидан хато ифодани аниқланг.

a)  $q = \% a + b + c + d / 4;$

b)  $c = (a \% b) * 6;$

c)  $d = c / b;$

d)  $e = (a + b + c + d) / 4;$

## Маъруза 5. Амаллар: инкремент, декремент, sizeof, мантиқий, разрядли, таққослаш. Амалларнинг устунликлари ва бажарилиш йўналишлари

### Режа:

1. Инкремент ва декремент амаллари
2. sizeof амали
3. Разрядли мантиқий амаллар
4. Чапга ва ўнгга суриш амаллари
5. Амалларнинг устунликлари ва бажарилиш йўналишлари

Таянч иборалар: *include, cout, cin, sizeof, Инкремент, Декремент, Префикс, Постфикс, Ўзгарувчи, Бит, Разряд.*

### 5.1. Инкремент ва декремент амаллари

C++ тилида операнд қийматини бирга ошириш ва камайтиришнинг самарали воситалари мавжуд. Булар инкремент (++) ва декремент (--) унар амаллардир.

```
soni=soni+1;
```

коди ўрнига

```
soni++;
```

кўринишида ёзилади.

Инкремент амали операнд қийматини биттага ошириш учун ишлатилади. Декремент амали эса операнд қийматини биттага камайтириш учун ишлатилади. Операндга нисбатан бу амалларнинг икки хил кўриниши: префикс ва постфикс кўринишлари мавжуд. Префикс кўринишда амал тил кўрсатмаси бўйича иш бажарилишидан олдин операндга қўлланилади. Постфикс ҳолатда эса амал тил кўрсатмаси бўйича иш бажарилгандан кейин операндга қўлланилади.

Префикс инкремент: `++variable`

Постфикс инкремент: `variable++`

Префикс декремент: `--variable`

Постфикс декремент: `variable--`

Префикс ёки постфикс амал тушунчаси фақат қиймат бериш билан боғлиқ ифодаларда ўринли.

$x = 5;$

$y = ++x;$

Ушбу мисолда  $x$  нинг қиймати беш.  $y$  га  $x$  нинг қийматини юклашдан олдин пре-инкремент амали қўлланилади,  $x$  нинг қиймати олтига ўзгаради, сўнгра  $y$  ўзгарувчисига олти юкланади. Натижада  $x$  нинг қиймати ҳам,  $y$  нинг қиймати ҳам олтига тенг бўлади.

$x = 5;$

$y = x++;$

Ушбу мисолда  $x$  нинг қиймати беш.  $y$  га  $x$  нинг қийматини юклаш жараёнида пост-инкремент амали қўлланилади,  $x$  нинг қиймати  $y$  ўзгарувчисига юкланади, сўнгра  $x$  нинг қиймати биттага оширилади. Натижада  $x$  нинг қиймати олти,  $y$  нинг қиймати бешга тенг бўлади.

Инкремент ва декремент амалларини мураккаб ифоданинг ичида ҳам ишлатиш мумкин. Фақат бунда, тушунарли бўлиши учун инкремент ва декремент амалларини қавс ичига олиш мақсадга мувофиқ.

$a = 5;$

$b = 2 + (++a);$

Биринчи ифодада  $a$  ўзгарувчисига беш сони юкланади. Кейин эса,  $2 + (++a)$  ифодани бажариш жараёнида аввал  $a$  ўзгарувчисининг қиймати биттага оширилади, сўнгра унинг қийматига икки сонини қўшиб, натижа  $b$  ўзгарувчисига юкланади. Натижада  $a$  нинг қиймати олти,  $b$  нинг қиймати саккизга тенг бўлади.

$a = 5;$

$b = 2 + (a--);$

Биринчи ифодада  $a$  ўзгарувчисига беш сони юкланади. Кейин эса,  $2 + (a--)$  ифодани бажариш жараёнида аввал  $a$  ўзгарувчисининг қиймати ифодага қўйиб ҳисобланиб, натижа  $b$  ўзгарувчисига юкланади, сўнгра унинг қиймати биттага камайтиради.. Натижада  $a$  нинг қиймати тўрт,  $b$  нинг қиймати эса еттига тенг бўлади.

## 5.2. sizeof амали

Ҳар хил турдаги ўзгарувчилар компьютер хотирасида турли сондаги байтларни эгаллайди. Бунда, ҳаттоки бир турдаги ўзгарувчилар ҳам қайси

компьютерда ёки қайси операцион системада амал қилинишига қараб турли ўлчамдаги хотирани банд қилиши мумкин.

C++ тилида ихтиёрий (таянч ва ҳосилавий) турдаги ўзгарувчиларнинг ўлчамини **sizeof** амали ёрдамида аниқланади. Бу амални ўзгармасга, турга ва ўзгарувчига қўлланиши мумкин.

Қуйида келтирилган дастурда компьютернинг платформасига мос равишда таянч турларининг ўлчамлари чоп қилинади.

```
#include <iostream>
using namespace std;
int main()
{
short s=-12;
double d=345.678;
unsigned long ul=123456789;
cout << "Size of char = " << sizeof(char) << endl;
cout << "Size of int = " << sizeof(int) << endl;
cout << "Size of short = " << sizeof(s) << endl;
cout << "Size of unsigned int = "
    << sizeof(unsigned int)<< endl;
cout << "Size of long = " << sizeof(long) << endl;
cout << "Size of bool = " << sizeof(bool) << endl;
cout << "Size of float = " << sizeof(float) << endl;
cout << "Size of double = " << sizeof(double) << endl;
cout << "Size of long double = " << sizeof(d) << endl;
cout << "Size of unsigned short = "
    << sizeof(unsigned short) << endl;
cout << "Size of unsigned long = "
    << sizeof(ul) << endl;
return 0;
}
```

Дастурнинг натижаси қуйидагича кўринишга эга:

```
Size of char = 1
Size of int = 4
Size of short = 2
```

Size of unsigned int = 4  
 Size of long = 4  
 Size of bool = 1  
 Size of float = 4  
 Size of double = 8  
 Size of long double = 8  
 Size of unsigned short = 2  
 Size of unsigned long = 4

### 5.3. Разрядли мантиқий амаллар

Дастур тузиш тажрибаси шуни кўрсатадики, одатда қўйилган масалани ечишда бирор ҳолат рўй берган ёки йўқлигини ифодалаш учун **0** ва **1** қиймат қабул қилувчи *байроқлардан* фойдаланилади. Бу мақсадда бир ёки ундан ортиқ байтли ўзгарувчилардан фойдаланиш мумкин. Масалан, боол туридаги ўзгарувчини шу мақсадда ишлатса бўлади. Бошқа томондан, байроқ сифатида байт-нинг разрядларидан фойдаланиш ҳам мумкин. Чунки разрядлар фақат иккита қийматни - **0** ва **1** сонларини қабул қилади. Бир байтда **8** разряд бўлгани учун унда **8** та байроқни кодлаш имконияти мавжуд.

Фараз қилайлик, қўриқлаш тизимига **5** та хона уланган ва тизим тахтасида **5** та чирокча (индикатор) хоналар ҳолатини билдиради: хона қўриқлаш тизими назоратида эканлигини мос индикаторнинг ёниб туриши (разряднинг **1** қиймати) ва хонани тизимга уланмаганлигини индикатор ўчганлиги (разряднинг **0** қиймати) билдиради. Тизим ҳолатини ифодалаш учун бир байт этарли бўлади ва унинг кичик разрядидан бошлаб бештасини шу мақсадда ишлатиш мумкин:

7	6	5	4	3	2	1	0
			<b>ind5</b>	<b>ind4</b>	<b>ind3</b>	<b>ind2</b>	<b>ind1</b>

Масалан, байтнинг қуйидаги ҳолати **1**, **4** ва **5** хоналар қўриқлаш тизимига уланганлигини билдиради:

7	6	5	4	3	2	1	0
<b>x</b>	<b>x</b>	<b>x</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>

Қуйидаги жадвалда С++ тилида байт разрядлари устида мантиқий амаллар мажмуаси келтирилган.

Байт разрядлари устида мантиқий амаллар

Амаллар	Мазмуни
&	Мантиқий ВА (қўпайтириш)
	Мантиқий ЁКИ (қўшиш)
^	Истисно қилувчи ЁКИ
~	Мантиқий ИНКОР (инверсия)

Разрядли мантиқий амалларнинг бажариш натижаларини жадвал кўринишида кўрсатиш мумкин.

Разрядли мантиқий амалларнинг бажариш натижалари

A	B	C=A&B	C=A B	C=A^B	C=~A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Юқоридаги келтирилган мисол учун кўриклаш тизимини ифодаловчи бир байтли чар туридаги ўзгарувчини эълон қилиш мумкин:

```
char q_taxtasi=0;
```

Бу ерда q\_taxtasi ўзгарувчисига 0 қиймат бериш орқали барча хоналар кўриклаш тизимига уланмаганлиги ифодаланади:

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Агар 3-хонани тизимга улаш зарур бўлса

```
q_taxtasi=q_taxtasi|0x04;
```

амалини бажариш керак, чунки  $0x04_{16}=00000100_2$  ва мантиқий ЁКИ амали натижасида q\_taxtasi ўзгарувчиси байти қуйидаги кўринишда бўлади:

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

Худди шундай йўл билан бошқа хоналарни тизимга улаш мумкин, зарур бўлса бирданига иккитасини (зарур бўлса барчасини):

```
q_taxtasi=q_taxtasi|0x1F;
```



Мантикий кўпайтириш орқали хоналарни кўриқлаш тизимидан чиқариш мумкин:

```
q_taxtasi=q_taxtasi&0xFD; // 0xFD16=11111102
```

Худди шу натижани Ъ~Ъ амалидан фойдаланган ҳолда ҳам олиш мумкин. Иккинчи хона тизимга уланганлиги билдирувчи байт қиймати - 00000010<sub>2</sub>, демак шу ҳолатни инкор қилган ҳолда мантикий кўпайтиришни бажариш керак.

```
q_taxtasi=q_taxtasi(~0x02);
```

Ва ниҳоят, агар 3-хона индикаторини, уни қандай қийматда бўлишидан қатъий назар қарама-қарши ҳолатга ўтказишни «инкор қилувчи ЁКИ» амали ёрдамида бажариш мумкин:

```
q_taxtasi=q_taxtasi^0x04; // 0x0416=000001002
```

Разрядли мантикий амалларни қиймат бериш оператори билан биргаликда бажарилишининг қуйидаги кўринишлари мавжуд:

**&=** – разрядли ВА қиймат бериш билан;

**|=** – разрядли ЁКИ қиймат бериш билан;

**^=** – разрядли истисно қилувчи ЁКИ қиймат бериш билан.

#### 5.4. Чапга ва ўнгга суриш амаллари

Байтдаги битлар қийматини чапга ёки ўнгга суриш учун, мос равишда “<<” ва “>>” амаллари қўлланилади. Амалдан кейинги сон битлар нечта ўрин чапга ёки ўнгга суриш кераклигини билдиради.

Масалан:

```
unsigned char A=12; // A=000011002=0x0C16
```

```
A=A<<2; // A=001100002=0x3016=4810
```

```
A=A>>3; // A=000001102=0x0616=610
```

Разрядларни  $n$  та чапга (ўнга) суриш сонни  $2^n$  сонига кўпайтириш (бўлиш) амали билан эквивалент бўлиб ва нисбатан тез бажарилади. Шунинг эътиборга олиш керакки, операнд ишорали сон бўлса, у ҳолда чапга суришда энг чапдаги ишора разряди такрорланади (ишора сақланиб қолади) ва манфий сонлар устида бу амал бажарилганда математика нуқтаи-назардан хато натижалар юзага келади:

```
char B=-120; // B=100010002=0x8816
```

$V = V \ll 2;$  //  $V = 00100000_2 = 0 \times 20_{16} = 32_{10}$   
 $V = -120;$  //  $V = 10001000_2 = 0 \times 88_{16}$   
 $V = V \gg 3;$  //  $V = 11110001_2 = 0 \times F1_{16} = -15_{10}$

Шу сабабли, бу разрядли суриш амаллари ишорасиз (unsigned) турдаги қийматлар устида бажарилгани маъкул.

Таққослаш амаллари.

C++ тилида қийматларни солиштириш учун таққослаш амаллари аниқланган. Таққослаш амали бинар амал бўлиб, қуйидаги кўринишга эга:

`<операнд1> <таққослаш амали> < операнд2>`

Таққослаш амалларининг натижаси - таққослаш ўринли бўлса, true (рост), акс ҳолда false (ёлғон) қиймат бўлади. Агар таққослашда арифметик ифода катнашса, унинг қиймати 0 қийматидан фарқли ҳолатлар учун 1 деб ҳисобланади.

Таққослаш амаллари ва уларнинг қўлланиши

Амаллар	Қўлланиши	Мазмуни (ўқилиши)
<	$a < b$	“a кичик b”
<=	$a \leq b$	“a кичик ёки тенг b”
>	$a > b$	“a катта b”
>=	$a \geq b$	“a катта ёки тенг b”
==	$a == b$	“a тенг b”
!=	$a != b$	“a тенг эмас b”

### 5.5. Амалларнинг устунликлари ва бажарилиш йўналишлари

Анъанавий арифметикадагидек C++ тилида ҳам амаллар маълум бир тартиб ва йўналишда бажарилади. Маълумки, математик ифодаларда бир хил устунликдаги (приоритетдаги) амаллар учраса (масалан, қўшиш ва айириш), улар чапдан ўнгга бажарилади. Бу тартиб C++ тилидаги ҳам ўринли, бироқ айрим ҳолларда амал ўнгдан чапга бажарилиши мумкин (хусусан, қиймат бериш амалида).

Ифодалар қийматини ҳисоблашда амаллар устунлиги ҳисобга олинади. Биринчи навбатда энг юқори устунликка эга бўлган амал бажарилади.

Қуйидаги жадвалда С++ тилида ишлатиладиган амаллар (операторлар), уларнинг устунлик коэффициентлари ва бажарилиш йўналишлари (← - ўнгдан чапга, ⇒ - чапдан ўнгга) келтирилган.

Амалларнинг устунликлари ва бажарилиш йўналишлари

Оператор	Тавсифи	Устунлик	Йўналиш
::	Кўриниш соҳасига рухсат бериш	16	⇒
[]	Массив индекси	16	⇒
()	Функцияни чақириш	16	⇒
.	Структура ёки синф элементини танлаш	16	⇒
->			
++	Постфикс инкремент	15	←
--	Постфикс декремент	15	←
++	Префикс инкремент	14	←
--	Префикс декремент	14	←
sizeof	Ўлчамни олиш	14	←
(<tur>)	Турга акслантириш	14	
~	Разрядли мантиқий ИНКОР	14	←
!	Мантиқий инкор	14	←
-	Унар минус	14	←
+	Унар плюс	14	←
&	Адресни олиш	14	←
*	Воситали мурожаат	14	←
new	Динамик объектни яратиш	14	←
delete	Динамик объектни йўқ қилиш	14	←
casting	Турга келтириш	14	
*	Кўпайтириш	13	⇒
/	Бўлиш	13	⇒
%	Бўлиш қолдиғи	13	⇒
+	Қўшиш	12	⇒
-	Айириш	12	⇒
>>	Разряд бўйича ўнгга суриш	11	⇒

<<	Разряд бўйича чапга суриш	11	⇒
<	Кичик	10	⇒
<=	Кичик ёки тенг	10	⇒
>	Катта	10	⇒
>=	Катта ёки тенг	10	⇒
==	Тенг	9	⇒
!=	Тенг эмас	9	⇒
&	Разрядли ВА	8	⇒
^	Разрядли истисно қилувчи ЁКИ	7	⇒
	Разрядли ЁКИ	6	⇒
&&	Мантиқий ВА	5	⇒
	Мантиқий ЁКИ	4	⇒
?:	Шарт амали	3	⇐
=	Қиймат бериш	2	⇐
*=	Кўпайтириш қиймат бериш билан	2	⇐
/=	Бўлиш қиймат бериш билан	2	⇐
%=	Модулли бўлиш қиймат бериш билан	2	⇐
+=	Қўшиш қиймат бериш билан	2	⇐
- =	Айириш қиймат бериш билан	2	⇐
<<=	Чапга суриш қиймат бериш билан	2	⇐
>>=	Ўнгга суриш қиймат бериш билан	2	⇐
&=	Разрядли ВА қиймат бериш билан	2	⇐
^=	Разрядли истисно қилувчи ЁКИ қиймат бериш билан	2	⇐
=	Разрядли ЁКИ қиймат бериш билан	2	⇐
throw	Истисно ҳолатни юзага келтириш	2	⇐
,	Вергул	1	⇒

C++ тили дастур тузувчисига амалларнинг бажарилиш тартибини ўзгартириш имкониятини беради. Худди математикадагидек, амалларни кавслар ёрдамида гуруҳларга жамлаш мумкин. Қавс ишлатишга чеклов йўқ.

Қуйидаги дастурда кавс ёрдамида амалларни бажариш тартибини ўзгартириш кўрсатилган.

```
#include <iostream>
using namespace std;
int main()
{
    int x=0, y=0;
    int a=3, b=34, c=82;
    x=a*b+c;
    y=(a*(b+c));
    cout<<"x= "<<x<<' \n' <<"y= "<<y<<' \n' ;
}
```

Дастурда амаллар устунлигига кўра **x** қийматини ҳисоблашда олдин **a** ўзгарувчи **b** ўзгарувчига кўпайтирилади ва унга **c** ўзгарувчи қийматига қўшилади. Навбатдаги кўрсатмани бажаришда эса биринчи навбатда ички кавс ичидаги ифода - **(b+c)** қиймати ҳисобланади, кейин бу қиймат **a** кўпайтирилиб, **y** ўзгарувчисига ўзлаштирилади. Дастур бажарилиши натижасида экранга

```
x=184
y=348
```

сатрлари чоп этилади.

#### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Мустақил таълим мавзулари

1. Сонларнинг компьютер хотирасида тасвирланиши.
2. Сонлар устида разрядли амаллар.
3. Сонлар устида таққослаш амаллари.
4. Санок системалари ва сонлар устида амаллар.

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
cout		экранга чиқариш оқими
cin		экрандан киритиш оқими
sizeof		ўзгарувчи турининг хотирадаги хажмини аниқлаш
Инкремент	Increment	ўзгарувчининг қийматини биттага ошириш
Декремент	Decrement	ўзгарувчининг қийматини биттага камайтириш
Префикс	Prefix	операторнинг ўзгарувчидан олдин жойлашган кўриниши
Постфикс	Postfix	операторнинг ўзгарувчидан кейин жойлашган кўриниши
Ўзгарувчи	variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
Бит	bit	энг кичик ўлчов бирлиги
Разряд	discharge	битлардан (0 ёки 1) ташкил топган индикатор

## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур координаталари билан берилган нуқтани чорагини аниқлаши керак. Лекин дастур нотўғри натижа бермоқда. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили да дастур кодини тақдим этинг (индивидуал ҳолда).

## Назорат саволлари

1. Инкремент ва декремент амаллари нима?
2. Префикс ёки постфикс амал тушунчаси қандай ифодаларда ўринли?
3. C++ тилида ихтиёрий (таянч ва ҳосилавий) турдаги ўзгарувчиларнинг ўлчамини қандай амали ёрдамида аниқланади?
4. Қўйилган масалани ечишда бирор ҳолат рўй берган ёки йўқлигини ифодалаш учун **0** ва **1** қиймат қабул қилувчи нималардан фойдаланилади?
5. C++ тилида байт разрядлари устида мантиқий амаллар мажмуаси жадвалини кўрсатинг.
6. 3-хона индикаторини, уни қандай қийматда бўлишидан қатъий назар карама-қарши ҳолатга ўтказишни қайси амал ёрдамида бажариш мумкин?
7. Байтдаги битлар қийматини чапга ёки ўнгга суриш учун, мос равишда қайси амаллари қўлланилади?
8. Таққослаш амали қандай амал бўлиб, у қандай кўринишга эга?
9. Таққослаш амалларининг натижаси - таққослаш ўринли бўлса ёки ўринли бўлмаса қандай қиймат бўлади?
10. Ифодалар қийматини ҳисоблашда нима ҳисобга олинади?

## Тест саволлари

1. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `int a=5; cout<< a++ <<endl;`
  - a) 5
  - b) 6
  - c) a++

d) 4

2. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `int a=5; int b=2+ (++a); cout<< b <<endl;`

a) 7

b) 7.0

c) 0.9

d) 7.9

3. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `int a=5; int b=a+ (++a); cout<< b <<endl;`

a) 12

b) 13

c) 11

d) 10

4. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `int a=5; int b=(a--) + (++a); cout<< b <<endl;`

a) 12

b) 11

c) 5

d) 6

5. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `int a=5; int b=(a--) + (++a); cout<< ++a <<endl;`

a) 6

b) 5

c) 4

d) 7

6. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `int a=5, b=2; b=(a>=b); cout << b << '\n';`

a) 1

b) 0

c) 5>=2

d) 2



7. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `int a=5, b=2; b=(a>=b); cout << b << '\n';`

- a) 1
- b) 0
- c) `5>=2`
- d) 2

8. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `bool c; int a=5, b=2; c=(a!=b) + (a>b); cout << c ;`

- a) 1
- b) 2
- c) 0
- d) 3

9. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `short a=5; unsigned int d=123; cout << sizeof(a) << ' ' << sizeof(d);`

- a) 2 4
- b) 4 2
- c) 5 123
- d) 2

10. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `short a=5; long d=123; cout << sizeof(d);`

- a) 4
- b) 2 4
- c) 5
- d) 123

## Маъруза 6. Ўқиш-ёзиш оқимлари (cin, cout)

### Режа:

1. Оқимлар
2. Ўқиш оқими (cin)
3. Ёзиш оқими (cout)

Таянч иборалар: *include, cout, cin, setw, ўзгарувчи, берилганлар, идентификатор, int, double, char.*

### 6.1. Оқимлар

Дастур учта асосий қисмдан ташкил топган, берилганларни ўқиш, берилганларни манипулятсия қилиш, натижаларни чиқариш. Олдинги мавзуларда сонли берилганларни бошқариш ва улар устида арифметик амаллар бажаришни кўрдик. Кейинги мавзуларимизда сонли бўлмаган берилганларни бошқариш ҳақида гапирилади. Модомики берилганларни ўқиш ва натижаларни чоп қилишда бир оз муаммога дуч келинадиган бўлса, C++да берилганларни ўқиш ва чоп қилишнинг кенг имкониятлари мавжуд.

C++ да берилганларни ўқиш ва чиқариш учун оқим деб номланувчи байтлар кетма-кетлиги мўлжалланган. Байтлар-бу белгилар кетма-кетлиги, агар дастур берилганларнинг бошқа турини (график турини) талаб этмаса. Шунинг учун оқиб берилганларнинг белгилар кетмас-кетлигини ўқиш учун мўлжалланган. Икки турдаги оқимлар мавжуд:

**Input stream:** иловадан берилганларнинг белгилар кетма-кетлигини киритиш;

**Output stream:** иловага берилганларнинг белгилар кетма-кетлигини чиқариш;

Стандарт ҳолда берилганлар дастур иловаси орқали клавиатурадан киритилади ва натижа дастур иловасига чиқарилади. Клавиатура орқали киритилаётган берилганлар кетма-кетлигини қабул қилиш ва экранга чиқариш учун, ҳар бир C++ дастур сарлавҳасида **iostream** файлдан фойдаланиши керак. **iostream** файли иккита оқимдан ташкил топган, **istream** – берилганларни киритиш оқими ва **ostream** – берилганларни чиқариш оқими. Шунингдек

сарлавха файли иккита оператордан **cin** -берилганларни оқимдан киритиш ва **cout** берилганларни оқимга чиқариш.

Бу операторлар ўзгарувчига ўхшайди ва қуйидагича ташкил топган:

```
istream cin;
```

```
ostream cout;
```

**cin** ва **cout** операторларидан фойдаланиш учун, ҳар бир C++ да тузилаётган дастур сарлавҳасида қуйидаги процессор директивасидан фойдаланиш керак:

```
#include <iostream>
```

## 6.2. Ўқиш оқими (cin)

Ўзгарувчиларнинг **istream** тури ўзгарувчиларнинг киритиш оқими дейилади; ўзгарувчиларнинг **ostream** тури ўзгарувчиларнинг чиқариш оқими дейилади. ўзгарувчиларнинг **astream** тури ўзгарувчиларнинг киритиш чиқаришнинг ихтиёрий оқими дейилади.

**cin** тури берилганларни **inputdevice** стандартдан олиш учун фойдаланиладиган операторлардан ва функциялардан фойдалана олиши мумкин. Сиз иловадан киритилган берилганларни олиш учун `>>` **brieflyused** операторидан фойдаланишингиз мумкин. Кейинги мавзуда `>>` операторидан қандай фойдаланиш мумкинлиги ҳамда ўзгарувчиларни киритишнинг **get**, **ignore**, **peek**, ва **putback** функцияларидан фойдаланишни ўрганасиз.

Берилганларни оқимдан киритиш учун **cin** калит сўзидан ва `>>` операторидан фойдаланилади. Ўзгарувчига киритиш оқимидан қиймат киритиш қуйидагича амалга оширилади:

```
cin >> soni;
```

Компилятор бу операторни бажараётганда киритиш оқимидан берилганни олиб хотирадаги **soni** ўзгарувчисида сақлайди. Шунинг учун фойдаланувчи клавиатурадан 15.50 қийматини киритса **soni** ўзгарувчисининг қиймати 15.50 га тенг бўлади;

Ўзгарувчиларни киритиш оператори `>>` иккита операнддан ташкил топган. Операторнинг чап томонида **cin** киритиш оқими ва операторнинг ўнг томонида ўзгарувчининг номи бўлади.

Берилганларни оқимдан ўқиш операторининг синтаксиси қуйидагича:

```
cin >> variable >> variable...;
```

Юқоридаги синтаксисдан кўришиб турибдики, битта ўзгарувчига киритиш оқимидан >> операторидан қайта-қайта фойдаланган ҳолда берилганларни ўқиб олиш мумкин. Бир нечта ўзгарувчига ҳам киритиш оқимидан берилганларни киритиш мумкин, қуйидагича:

```
cin >> soni >> vazni;
```

шунингдек, юқоридаги операторни қуйидагича ёзиш мумкин:

```
cin >> soni;
```

```
cin >> vazni;
```

Бу мисолда берилганлар оқимидан олдин **soni** ўзгарувчисига қиймат киритилади ва янги қаторга тушганда **vazni** ўзгарувчисига қиймат киритилади ёки аксинча қора экранда битта қаторда киритилаётган қийматлар пробел орқали киритилади.

```
cin >> soni >> vazni;
```

Қуйидаги кўринишда қийматлар киритилиши мумкин:

```
15 48.30
```

ёки:

```
15
```

```
48.30
```

Киритиш оператори **soni** ўзгарувчисига 15 қийматини ва **vazni** ўзгарувчисига 48.30 қийматларини ўзлаштиради

Ўзгарувчига берилиши мумкин бўлган қийматлар қуйидаги жадвалда келтирилган:

Ўзгарувчининг тури	Қабул қилиш қиймати
char	Пробелдан бошқа битта белгини қабул қилади
int	Бутун турдаги ихтиёрий қийматни қабул қилади
double	Ҳақиқий турдаги ихтиёрий қийматларни қабул қилади

Бизга ўзгарувчиларнинг тури қуйидагича эълон қилинган бўлсин

```
int a, b;
```

```
double z;
```

```
char ch;
```

Ўзгарувчиларни **cin** калит сўзи ёрдамида >> операторидан фойдаланиб ўқиш ва қийматларни консол иловадан киритиш қуйидагича амалга оширилади:

№	Қийматларни киритиш	Қийматларни киритиш тартиби кўринишлари	Хотирадаги ўзгарувчилар қийматлари
1.	cin >> ch;	A	ch='A'
2.	cin >> ch;	AB	ch='A', 'B'
3.	cin >> a;	48	a=48
4.	cin >> a;	46.35	a=46, .35
5.	cin >> z;	74.35	z=74.35
6.	cin >> z;	39	z=39.0
7.	cin >> z >> a;	65.78 38	z=65.78, a=38
8.	cin >> a >> b;	4 60	a=4, b=60
9.	cin >> a >> z;	46 32.4 68	a=46, z=35.4, 68

Агар бизга ўзгарувчиларнинг тури қуйидагича эълон қилинган бўлса:

```
int a;
```

```
double z;
```

```
char ch;
```

Ўзгарувчиларни **cin** калит сўзи ёрдамида >> операторидан фойдаланиб ўқиш ва қийматларни консол иловадан киритиш қуйидагича амалга оширилади:

№	Қийматларни киритиш	Қийматларни киритиш тартиби кўринишлари	Хотирадаги ўзгарувчилар қийматлари
1.	<code>cin &gt;&gt; a &gt;&gt; ch &gt;&gt; z;</code>	57 A 26.9	a=57, ch='A', z=26.9
2.	<code>cin &gt;&gt; a &gt;&gt; ch &gt;&gt; z;</code>	57 A 26.9	a=57, ch='A', z=26.9
3.	<code>cin &gt;&gt; a &gt;&gt; ch &gt;&gt; z;</code>	57 A 26.9	a=57, ch='A', z=26.9
4.	<code>cin &gt;&gt; a &gt;&gt; ch &gt;&gt; z;</code>	57A26.9	a=57, ch='A', z=26.9

Кўришиб турибтики юқоридаги 1-4 ҳолларда қийматларни ўқиб олиш бир хил фақатгина қиймат киритиш ҳар хил. 1-ҳолда қийматлар битта қаторда пробел ёрдамида ажратиб киритилияпти; 2-ҳолда қийматлар иккита қаторда киритилияпти, улардан биринчи иккитаси пробел билан ажратиб битта қаторда учинчи қиймат янги қатордан киритилияпти; 3- ҳолда барча қийматлар янги қатордан киритилияпти; 4-ҳолда берча қийматлар пробел билан ажратилмай кўшиб киритилияпти. Қийматлардан билиш мумкинки иккинчи қиймат сон эмас, бу қиймат ўқиб олишлар қуйидагича кўринишда ишляпти: 1, 2, 3 ҳоллар деярли бир хил тушиниш осон, 4-ҳолда >> оператори аввал 57 сонини оқимдан ажратиб олиб а га берди ва белги учраганда уни белги туридаги ч ўзгарувчисига берди, 26.9 ни з ўзгарувчисига берди.

Агар бизга ўзгарувчиларнинг тури қуйидагича эълон қилинган бўлса:

```
int a , b;
double z;
char ch, ch1, ch2;
```

Ўзгарувчиларни **cin** калит сўзи ёрдамида >> операторидан фойдаланиб ўқиш ва қийматларни консол иловадан киритиш қуйидагича амалга оширилади:

№	Қийматларни киритиш	Қийматларни киритиш тартиби кўринишлари	Хотирадаги ўзгарувчилар қийматлари
1.	<code>cin &gt;&gt; z &gt;&gt; ch &gt;&gt; a;</code>	36.78B34	<code>z=36.78</code> <code>ch='B'</code> , <code>a=34</code>
2.	<code>cin &gt;&gt; z &gt;&gt; ch &gt;&gt; a;</code>	36.78 B34	<code>z=36.78</code> <code>ch='B'</code> , <code>a=34</code>
3.	<code>cin &gt;&gt; z &gt;&gt; ch &gt;&gt; a;</code>	11 34	<code>a=11</code> , <code>b=34</code>
4.	<code>cin &gt;&gt; a &gt;&gt; z;</code>	78.49	<code>a=78</code> <code>z=0.49</code>
5.	<code>cin &gt;&gt; ch &gt;&gt; a;</code>	256	<code>ch='2'</code> , <code>a=56</code>
6.	<code>cin &gt;&gt; a &gt;&gt; ch;</code>	256	<code>a=256</code> <code>kompyuter</code> <code>ch</code> <code>o'zgaruvchisi</code> <code>qiymati</code> <code>kiritilishini</code> <code>kutadi</code>
7.	<code>cin &gt;&gt; ch1 &gt;&gt; ch2 &gt;&gt; a;</code>	A B	<code>ch1='A'</code> , <code>ch='B'</code>

Қуйида мисол сифатида келтирилган дастурда, дастур кодидан олдин сарлавҳа файллари аниқлаб олишган. Биз бин нечта математик амаллардан фойдаланишимиз учун **cmath** сарлавҳаси ва сатрлар устида амаллаб бажариш учун `string` сарлавҳа файли қўшилган, `length` функцияси `string` туридаги сатрнинг узунлигини аниқлаб беради:

```
#include <iostream>
#include <cmath>
#include <string>
using namespace std;
int main()
```

```

{
double u, v;
string str;
cout << "Line 1: 2 to the power of 6 = "
      << static_cast<int>(pow(2.0, 6.0)) << endl;

u = 12.5;
v = 3.0;
cout << "Line 4: " << u << " to the power of "
      << v << " = " << pow(u, v) << endl;
cout << "Line 5: Square root of 24 = "
      << sqrt(24.0) << endl;
u = pow(8.0, 2.5);
cout << "Line 7: u = " << u << endl;
str = "Programming with C++";
cout << "Line 9: Length of str = "
      << str.length() << endl;
return 0;
}

```

Дастур натижаси:

```

Line 1: 2 to the power of 6 = 64
Line 4: 12.5 to the power of 3 = 1953.13
Line 5: Square root of 24 = 4.89898
Line 7: u = 181.019
Line 9: Length of str = 20

```

### **cin оператори ва get функцияси**

Қиймат дастур орқали ўқиб олинганида пробеллар қийматларнинг ажратувчиси сифатида қабул қилинади. Агар пробелнинг ўзи қиймат сифатида олиниши керак бўлса get функциясидан фойдаланиш керак:

```

char ch1, ch2;
int num;

```

ва қуйидаги қийматлар киритилсин



A 25

Қийматларни киритиб олайлик:

```
cin >> ch1 >> ch2 >> num;
```

бу оператор бажарилаётганда ‘А’ қийматни ch1 ўзгарувчисига пробел белгиси ташлаб юборилади ва ‘2’ қийматини ch2 ўзгарувчиси ва 5 қийматини num ўзгарувчиси қабул қилади. Агарда сиз пробел белгисини ch1 ўзгарувчисига ва 25 қийматини ch2 ўзгарувчисига ўзлаштирмоқчи бўлсангиз, кўриниб турибтики бунда сиз юқоридаги қийматларни ўзлаштиришда >> киритиш операторидан фойдалана олмайсиз.

cin калит сўзи орқали киритиш оқимидаги бир нечта функцияларда фойдалана олишингиз мумкин. Белгилар кетма-кетлигини ўқиб олиш учун **get** функциясидан фойдаланиш мумкин, унинг структураси қуйидагича:

```
cin.get(varchar);
```

Мисол учун:

```
cin.get(ch1);
```

```
cin.get(ch2);
```

```
cin >> num;
```

қуйидаги қийматлар киритилади:

A 25

А қийматни ch1 ўзгарувчиси, пробел белгисини ch2 ўзгарувчиси ва 25 сонини num ўзгарувчиси қабул қилади.

cin.get()- функцияси белги туридаги ўзгарувчига фақат битта белгини киритиш учун мўлжалланган. Агарда бир нечта белгини киритилиши керак бўлса, get функцияси бир неча марта чақирилади.

### **cin оператори ва ignore функцияси**

Агарда берилганлар оқимидан фақат керакли қисмини киритиш керак бўлса, унда киритиш оқимининг ignore функциясидан фойдаланиш мумкин. ignore функциясининг синтаксиси қуйидагича:

```
cin.ignore(inexp, chexp);
```

`ignore` функцияси иккита параметрдан ташкил топган бўлиб, биринчи параметер `int` турида, иккинчи параметри `char` белги турида, мисол тариқасида қуйидаги дастурни кўрайлик:

Ўзгарувчилар қиймати қуйидагича бўлсин

```
int a, b;
cin >> a;
cin.ignore(100, '\n');
cin >> b;
```

Қуйидаги қийматлар киритилсин:

```
25 67 89 43 72
12 78 34
```

Бу ерда Киритиш оқими 25 қийматини `a` ўзгарувчисига ўқийди Иккинчи оператор `cin.ignore(100, '\n');`, `'\n'` – белгисигача инкор қилади ва `cin >> b`, киритиш оператори 12 қийматини `b` ўзгарувчисига ўқиб олади.

### 6.3. Ёзиш оқими (`cout`)

Берилганларни оқимга чиқариш учун `cout` калит сўзидан ва `<<` операторидан фойдаланилади.

```
cout << soni;
```

Бир нечта ўзгарувчига ҳам киритиш оқимидан берилганларни киритиш мумкин, қуйидагича:

```
cout << soni << vazni;
```

шунингдек, юқоридаги операторни қуйидагича ёзиш мумкин:

```
cout << soni;
```

```
cout << vazni;
```

Берилганларни турли форматда ва кўринишда чоп этиш учун манипуляторлардан фойдаланилади. Манипуляторларни дастурда ишлатиш учун `iomanip` кутубхонасидан фойдаланиш керак.

`fixed` манипулятори ҳақиқий сонни фиксирланган нуқтали кўринишда чоп этади. Ушбу манипулятордан фойдаланиш имконини `cout.unsetf(ios::fixed);` функцияси орқали ўчириб қўйиш мумкин. `scientific` манипулятори эса ҳақиқий сонни илмий форматда (экспоненциал) чоп этишда ишлатилади.

```
#include <iostream>
using namespace std;
int main()
{
double hours = 35.45;
double rate = 15.00;
double tolerance = 0.01000;
cout << "hours = " << hours << ", rate = "
    << rate << ", pay = " << hours * rate
    << ", tolerance = " << tolerance << endl << endl;
cout << scientific;
cout << "Scientific notation: " << endl;
cout << "hours = " << hours << ", rate = " << rate
    << ", pay = " << hours * rate
    << ", tolerance = " << tolerance << endl << endl;
cout << fixed;
cout << "Fixed decimal notation: " << endl;
cout << "hours = " << hours << ", rate = " << rate
    << ", pay = " << hours * rate
    << ", tolerance = " << tolerance << endl << endl;
return 0;
}
```

Дастур натижаси:

```
hours = 35.45, rate = 15, pay = 531.75, tolerance = 0.01
Scientific notation:
hours = 3.545000e+001, rate = 1.500000e+001, pay =
5.317500e+002, tolerance = 1.000000e-002
Fixed decimal notation:
```

```
hours = 35.450000, rate = 15.000000, pay = 531.750000,  
tolerance = 0.010000
```

`setprecision` манипулятори хақиқий сонларни чоп этишда ишлатилади. Бу манипулятор орқали сон каср қисмининг нечта рақамини чоп этиш кераклагини аниқлаш имкони туғилади.

```
cout << setprecision(2);  
double d=123.456;  
cout << fixed << setprecision(2);  
cout << d;
```

Дастур натижаси: 123.45 сони экранга чоп этилади.

`setw` манипулятори ўзгарувчи ёки ифода қиймати натижаларини махсус катакларда (жой) чиқариш имконини беради. `setw(n)` – кўринишида бериладиган ушбу манипуляторда `n` – нечта катакчада чиқариш кераклигини аниқлайди.

```
#include <iostream>  
using namespace std;  
int main()  
{  
int x = 19;  
int a = 345;  
double y = 76.384;  
cout << fixed << showpoint;  
cout << "12345678901234567890" << endl;  
cout << setw(5) << x << endl;  
cout << setw(5) << a << setw(5) << "Hi"  
<< setw(5) << x << endl << endl;  
cout << setprecision(2);  
cout << setw(6) << a << setw(6) << y  
<< setw(6) << x << endl;  
cout << setw(6) << x << setw(6) << a  
<< setw(6) << y << endl << endl;  
cout << setw(5) << a << x << endl;
```

```
cout << setw(2) << a << setw(4) << x << endl;
return 0;
}
```

Дастур натижаси:

12345678901234567890

19

345 Hi 19

345 76.38 19

19 345 76.38

34519

345 19

#### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

#### Мустақил таълим мавзулари

1. Ўқиш оқими сифатида файллардан фойдаланиш.
2. Ёзиш оқими сифатида файллардан фойдаланиш.
3. Ёзиш оқимида манипуляторлардан фойдаланиш.
4. <conio.h> кутубхонаси орқали аниқланган киритиш функциялари.

## Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
cout		экранга чиқариш оқими
cin		экрандан киритиш оқими
setw		ўзгарувчининг белги билан тўлдириб чиқариш
ўзгарувчи	variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
берилганлар	data	дастур ишлаши учун керакли қийматлар
идентификатор	identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизиқ ('_') белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги
int		бутун сон кўринишидаги берилганларнинг тури
double		ҳақиқий сон кўринишидаги берилганларнинг тури
char		белги кўринишидаги берилганларнинг тури

## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур экрандан киритилган сонни ўн та белги узунлигига ўнг томондан жойлаштириши ва бўш жойларни “!” белгиси билан тўлдириб чиқиши керак. Дастур сон узунлиги фақат икки хонали бўлганда ишламоқда. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

## Назорат саволлари

1. Ўқиш оқими нима?
2. Байтлар-бу....
3. Input stream нима?
4. Output stream нима?
5. C++ дастур сарлавҳасида қайси файлдан фойдаланиш керак?
6. iostream файли нечта оқимдан ташкил топган?
7. Қиймат дастур орқали ўқиб олинганида нималар қийматларнинг ажратувчиси сифатида қабул қилинади?
8. Қайси калит сўзи орқали киритиш оқимидаги бир нечта функцияларда фойдаланиш мумкин?
9. Берилганлар оқимидан фақат керакли қисмини киритиш керак бўлса, унда киритиш оқимининг қайси функциясидан фойдаланиш керак?
10. `fixed` манипулятори нимани чоп этади?

## Тест саволлари

1. C++ тилида консол режимида ишлаш жараёнида берилганларни стандарт оқимдан о'қиб форматига то'ғ'ри кўрсатилган жавобни аниқланг.
  - a) `cin>> <o'zgaruvchi>`
  - b) `cout<< <ifoda>`
  - c) `cin<< <o'zgaruvchi>`
  - d) `cout>><o'zgaruvchi>`

2. C++ tilida konsol rejimda ishlash jarayonida ma'lumotlarni standart oqimga (ekranga)chiqarish uchun to'g'ri format keltirilgan javobni toping.

- a) `cout<< <ifoda>`
- b) `cin>> <o'zgaruvchi>`
- c) `cout>> <ifoda>`
- d) To'g'ri javob keltirilmagan

3. Quyidagi programma ishlashi natijasida ekranda qanday qiymat aks etadi?

```
#include <iostream.h>
int main () {
int a=10, b;
b=a%10;
cout<<b;
return 0; }
```

- a) \*0
- b) 1
- c) 10
- d) 100

4. Quyidagi programma ishlashi natijasida ekranda qanday qiymat aks etadi?

```
#include <iostream.h>
int main () {
int a=10, b;
if (a%2==0) b=a*a;
else
b=-a;
cout<<b;
return 0; }
```

- a) \*100
- b) -10
- c) 10
- d) 0



5. Quyidagi programma ishlashi natijasida ekranda qanday qiymat aks etadi?

```
#include <iostream.h>
int main () {
int a=10, b=23;
switch(a) {
case 1: c=a+b; break;
case 2: c=a*b; break;
default: c=(a+b)*b; break;
}
cout<<c;
return 0; }
```

- a) \*759
- b) 33
- c) 230
- d) xatolik haqida habar beradi

6. Oqimdan satrni o'qishga mo'ljallangan gets() funksiyasining prototipi to'g'ri ko'rsatilgan javobni aniqlang.

- a) \*char\* gets(char \*s);
- b) char gets(char \*s);
- c) char\* gets(char s);
- d) string\* gets(string \*s);

7. C++ tilida belgi turidagi o'zgaruvchilar qaysi kalit so'z yordamida anilanadi?

- a) \*char
- b) string
- c) double
- d) int

8. C++ tilida belgi turidagi o'zgaruvchilar qaysi kalit so'z yordamida anilanadi?

- a) \*char
- b) string
- c) double
- d) int

9. C++ tilida char turidagi o'zgaruvchilarning qabul qiladigan qiymatlar oralig'i to'g'ri ko'rsatilgan javobni aniqlang.

- a) \*0..255
- b) -128..127
- c) 0..32
- d) -32768..32767

10. Agar o'qsih muvafaqiyatli bo'lsa getc() funksiyasi qanday qiymat qaytaradi?

- a) \*ishorasiz int ko'rinishidagi qiymatni
- b) EOF ni qaytaradi
- c) ishorasli int ko'rinishidagi qiymatni
- d) Haqiqiy sonni

## Маъруза 7-8. Операторлар. Шарт операторлари

### Режа:

1. Таққослаш амаллари
2. Мантиқий операторлар
3. Шарт операторлари. **if** оператори
4. **if...else** оператори
5. **?:** шарт амали
6. **switch** оператори

Таянч иборалар: *include, cout, cin, if, ўзгарувчи, идентификатор, else, switch, case, default.*

### 7.1. Таққослаш амаллари

C++ тилида қийматларни солиштириш учун таққослаш амаллари аниқланган. Таққослаш амали бинар амал бўлиб, қуйидаги кўринишга эга:

**<операнд1> <таққослаш амали> < операнд2>**

Таққослаш амалларининг натижаси - таққослаш ўринли бўлса, true (рост), акс ҳолда false (ёлғон) қиймат бўлади. Агар таққослашда арифметик ифода қатнашса, унинг қиймати 0 қийматидан фарқли ҳолатлар учун 1 деб ҳисобланади.

#### Таққослаш амаллари ва уларнинг қўлланиши

Амаллар	Қўлланиши	Мазмуни (ўқилиши)
<b>&lt;</b>	<b>a&lt;b</b>	“a кичик b”
<b>&lt;=</b>	<b>a&lt;=b</b>	“a кичик ёки тенг b”
<b>&gt;</b>	<b>a&gt;b</b>	“a катта b”
<b>&gt;=</b>	<b>a&gt;=b</b>	“a катта ёки тенг b”
<b>==</b>	<b>a==b</b>	“a тенг b”
<b>!=</b>	<b>a!=b</b>	“a тенг эмас b”

## 7.2. Мантиқий операторлар

Дастурлашда бир эмас балки бир нечта шартли ифодаларни текшириш зарурияти жуда кўп учрайди. Масалан,  $x$  ўзгарувчиси  $y$  ўзгарувчисидан,  $y$  эса ўз навбатида  $z$  ўзгарувчисидан каттами шarti бунга мисол бўла олади. Бизнинг дастуримиз мос амални бажаришдан олдин бу иккала шарт рост ёки ёлғонлигини текшириши лозим.

Қуйидаги мантиқ асосида юқори даражада ташкил қилинган сигнализация системасини тасаввур қилинг. Агарда эшикда сигнализация ўрнатилган бўлса ВА кун вақти кеч соат олти ВА бугун байрам ЁКИ дам олиш куни БЎЛМАСА полиция чақирилсин. Барча шартларни текшириш учун C++ тилининг учта мантиқий оператори ишлатилади.

### Мантиқий операторлар

Оператор	Белги	Намуна
ВА	&&	ифода1 && ифода2
ЁКИ		ифода1    ифода2
ИНКОР	!	!ифода

Мантиқий кўпайтириш оператори иккита ифодани ҳисоблайди, агар иккала ифода `true` қиймат қайтарса ВА оператори ҳам `true` қиймат қайтарди. Агарда сизнинг қорнингиз очлиги рост бўлса ВА сизда пул борлиги ҳам рост бўлса сиз супермаркетга боришингиз ва у ердан ўзингизга тушлик қилиш учун бирор бир нарса харид қилишингиз мумкин. Ёки яна бир мисол, масалан,

$(x==5) \ \&\& \ (y==5)$

мантиқий ифодаси агарда  $x$  ва  $y$  ўзгарувчиларини иккаласининг ҳам қийматлари 5 га тенг бўлсагина `true` қиймат қайтаради. Бу ифода агарда ўзгарувчилардан бирортаси 5 га тенг бўлмаган қиймат қабул қилса `false` қийматини қайтаради. Мантиқий кўпайтириш оператори фақатгина ўзининг иккала ифодаси ҳам рост бўлсагина `true` қиймат қайтаради.

Мантиқий кўпайтириш оператори `&&` белги орқали белгиланади.

ифода1	ифода1	ифода1 && ифода2
false (0)	false (0)	false (0)
false (0)	true (0 emas)	false (0)
true (0 emas)	false (0)	false (0)
true (0 emas)	true (0 emas)	true (1)

Мантикий кўшиш оператори ҳам иккита ифода орқали ҳисобланади. Агарда улардан бирортаси рост бўлса мантикий кўшиш оператори true қиймат қайтаради. Агарда сизда пул ёки кредит карточкаси бўлса, сиз счётни тўлай оласиз. Бу ҳолда иккита шартнинг бирданига бажарилиши: пулга ҳам ва кредит карточкасига ҳам эга бўлишингиз шарт эмас. Сизга улардан бирини бажарилиши етарли. Бу операторга оид яна бир мисолни қараймиз. Масалан,

`(x==5) || (y>13)`

ифодаси ёки x ўзгарувчи қиймати 5 га тенг бўлса, ёки y ўзгарувчи қиймати 13 дан катта бўлса рост қиймат қайтаради.

Мантикий кўшиш оператори || белги орқали белгиланади.

ифода1	ифода1	ифода1    ифода2
false (0)	false (0)	false (0)
false (0)	true (0 emas)	true (1)
true (0 emas)	false (0)	true (1)
true (0 emas)	true (0 emas)	true (1)

Мантикий инкор оператори текширилатган ифода ёлгон бўлса true қиймат қайтаради. Агарда текширилатган ифода рост бўлса инкор оператори false қиймат қайтаради. Масалан,

`!('A' > 'B')` ёки `!(x <= 9)`

Мантикий инкор оператори ! белги орқали белгиланади.

ифода1	!ифода1
false (0)	true (1)
true (0 emas)	false (0)

### 7.3. Шарт операторлари

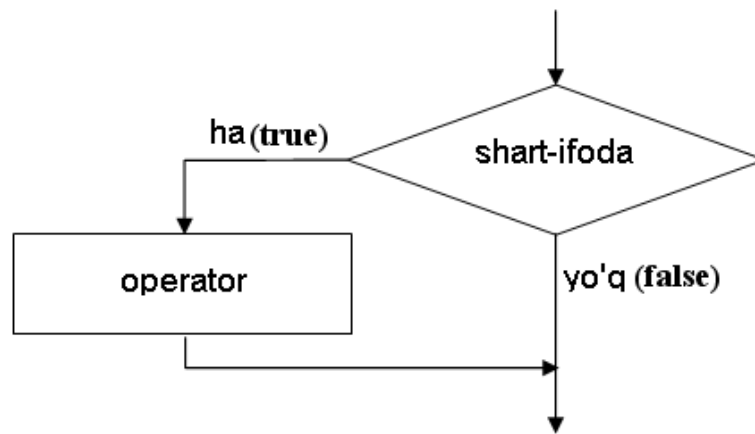
Олдинги бобда мисол тариқасида келтирилган дастурларда амаллар ёзилиш тартибида кетма-кет ва фақат бир марта бажариладиган ҳолатлар, яъни чизиқли алгоритмлар келтирилган. Амалда эса камдан-кам масалалар шу тариқа ечилиши мумкин. Аксарият масалалар юзага келадиган турли ҳолатларга боғлиқ равишда мос қарор қабул қилишни (ечимни) талаб этади. C++ тили дастурнинг алоҳида бўлақларининг бажарилиш тартибини бошқаришга имкон берувчи қурилмаларнинг етарлича катта мажмуасига эга. Масалан, дастур бажарилишининг бирорта қадамида қандайдир шартни текшириш натижасига кўра бошқарувни дастурнинг у ёки бу бўлагига узатиш мумкин (тармоқланувчи алгоритм). Тармоқланишни амалга ошириш учун шартли оператордан фойдаланилади.

**if** оператори қандайдир шартни ростликка текшириш натижасига кўра дастурда тармоқланишни амалга оширади:

**if** (<шарт>) <оператор>;

Бу ерда <шарт> ҳар қандай ифода бўлиши мумкин, одатда у таққослаш амали бўлади.

Агар шарт 0 қийматидан фарқли ёки рост (**true**) бўлса, <оператор> бажарилади, акс ҳолда, яъни шарт 0 ёки ёлғон (**false**) бўлса, ҳеч қандай амал бажарилмайди ва бошқарув **if** операторидан кейинги операторга ўтади (агар у мавжуд бўлса). Бундай конструкция бир томонлама танлов деб ҳам аталади. Ушбу ҳолат қуйидаги расмда кўрсатилган.



**if** шарт операторининг бажарилиши (бир томонлама танлов)

Қуйида келтирилган дастурда **if** операторидан фойдаланиш кўрсатилган.

```

#include <iostream>
using namespace std;
int main()
{
  int b;
  cin >> b;
  if (b > 0)
    cout << "b - musbat son";
  b = b - 4;
  if (b < 0)
    cout << "b - manfiy son";
  return 0;
}
  
```

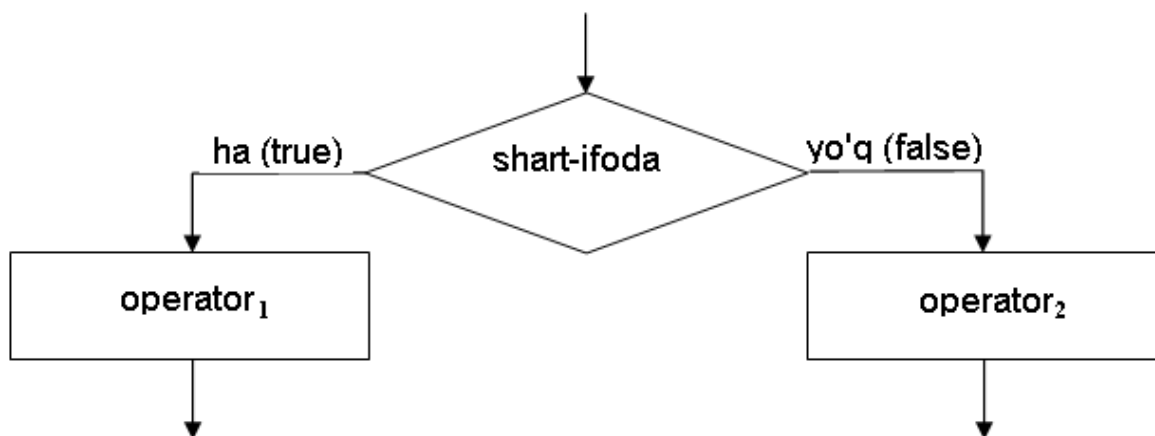
Дастур бажарилиши жараёнида бутун турдаги **b** ўзгарувчи эълон қилинади ва унинг қиймати клавиатурадан ўқилади. Кейин **b** қийматини 0 сонидан катталиги текширилади, агар шарт бажарилса (**true**) , у ҳолда экранга "**b - musbat son**" хабари чиқади. Агар шарт бажарилмаса, бу операторлар чеклаб ўтилади. **b** ўзгарувчисининг қийматидан тўрт айирилади. Навбатдаги шарт оператори **b** ўзгарувчи қиймати манфийликка текширади, агар шарт бажарилса, экранга "**b - manfiy son**" хабари чиқади.

## 7.4. if...else оператори

Агар дастур бажарилиши жараёнида шартнинг натижасига қараб у ёки бу амални бажариш керак бўлса, шарт операторининг икки томонлама танловли кўринишидан фойдаланилади. Шарт операторининг икки томонлама танловли кўриниши - **if...else** куйидагича синтаксисга эга:

```
if (<shart-ifoda>)  
    <operator1>;  
else  
    <operator2>;
```

Бу ерда <шарт-ифода> 0 қийматидан фаркли ёки true бўлса, <оператор<sub>1</sub>>, акс ҳолда <оператор<sub>2</sub>> бажарилади. **if...else** шарт оператори мазмунига кўра алгоритмнинг тармоқланувчи блокини ифодалайди: <шарт-ифода> - шарт блоки (ромб) ва <оператор<sub>1</sub>> блокнинг «ҳа» шохига, <оператор<sub>2</sub>> эса блокнинг «йўқ» шохига мос келувчи амаллар блоклари деб караш мумкин.



**if...else** шарт операторининг бажарилиши  
(икки томонлама танлов)

Қуйида келтирилган дастурда **if...else** операторидан фойдаланиш кўрсатилган.

```
#include <iostream>  
using namespace std;  
int main()
```



```

{
int a, b, c;
cin >> a >> b;
if (a + b > 30)
    c = a + b;
else
    c = a * b;
cout << "c = " << c;
return 0;
}

```

Дастур бажарилиши жараёнида бутун турдаги **a** ва **b** ўзгарувчиларига мос равишда 9 ва 13 сонлари киритилсин. Шарт текширилиши жараёнида **a** ва **b** ўзгарувчилари қийматлари кўшилади ва ўттиз сонидан катталиги текширилади. Агар натижа ўттиздан катта бўлса **c** ўзгарувчисига **a** ва **b** ўзгарувчилари қийматлари йиғиндиси юкланади, акс ҳолда уларнинг кўпайтмаси юкланади. Экранга "c = 117" хабари чиқади.

C++ тилининг қурилмалари операторларни блок кўринишида ташкил қилишга имкон беради. *Блок* – ‘{’ ва ‘}’ белгилари оралиғига олинган операторлар кетма-кетлиги бўлиб, у компилятор томонидан яхлит бир оператор деб қабул қилинади. Блок ичида эълон операторлари ҳам бўлиши мумкин ва уларда эълон қилинган ўзгарувчилар фақат шу блок ичида кўринади (амал қилади), блокдан ташқарида кўринмайди. Блокдан кейин ‘;’ белгиси қўйилмаслиги мумкин, лекин блок ичидаги ҳар бир ифода ‘;’ белгиси билан яқунланиши шарт.

```

{
    ифода_1
    ифода_2
    .
    .
    .
    ифода_n
}

```

Масала. Берилган тўрт хонали ишорасиз соннинг бошидаги иккита рақамининг йиғиндиси қолган рақамлар йиғиндисига тенг ёки йўқлиги аниқлансин (рақамлар йиғиндиси деганда уларга мос сон қийматларининг йиғиндиси тушунилади). Соннинг рақамларини ажратиб олиш учун бутун сонлар арифметикаси амалларидан фойдаланилади:

```
#include <iostream>
using namespace std;
int main()
{
    unsigned int n, a3, a2, a1, a0;
    cout << "\nn - qiymatini kiriting: ";
    cin >> n;
    if ((n < 1000) || (n > 9999))
    {
        cout << "Kiritilgan son 4 xonali emas!";
    }
    else
    {
        a3 = n / 1000;
        a2 = n % 1000 / 100;
        a1 = n % 100 / 10;
        a0 = n % 10;
        if (a3 + a2 == a1 + a0) cout << "a3+a2 = a1+a0";
        else cout << "a3+a2 != a1+a0";
    }
    return 0;
}
```

Дастур ишорасиз бутун сон киритишни таклиф қилади. Агар киритилган сон **4** хонали бўлмаса ( $n < 1000$ ) ёки ( $n > 9999$ ), бу ҳақда хабар берилади ва дастур ўз ишини тугатади. Акс ҳолда  $n$  сонининг рақамлари ажратиб олинади, ҳамда бошидаги иккита рақамнинг йиғиндиси -  $(a3+a2)$  қолган иккита рақамлар йиғиндиси -  $(a1+a0)$  билан солиштирилади ва уларнинг тенг ёки йўқлиги қараб мос жавоб чоп қилинади.

Шарт операторида эълон қилиш операторларини ишлатиш ман этилади, лекин ундаги блокларда ўзгарувчиларни эълон қилиш мумкин ва бу ўзгарувчилар фақат блок ичида амал қилади. Қуйидаги мисолда бу ҳолат билан боғлиқ хатолик кўрсатилган:

```
if (j > 0)
{
    int i;
    i = 2 * j;
}
else i = -j;
// хато, chunki i blokdan tashqarida ko'rinmaydi
```

Мисол тариқасида дискриминантни ҳисоблаш усули ёрдамида  $ax^2+bx+c=0$  кўринишидаги квадрат тенглама илдизларини топиш масаласини кўрайлик:

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    float a, b, c;
    float D, x1, x2;
    cout << "ax^2+bx+c=0 tenglama ildizini topish. ";
    cout << "\n a - koeffisiyentini kiriting: ";
    cin >> a;
    cout << "\n b - koeffisiyentini kiriting: ";
    cin >> b;
    cout << "\n c - koeffisiyentini kiriting: ";
    cin >> c;
    D = b * b - 4 * a * c;
    if (D < 0)
    {
        cout << "Tenglama haqiqiy ildizga ega emas!";
        return 0;
    }
}
```

```

if (D == 0)
{
    cout << "Tenglama yagona ildizga ega: ";
    x1 = -b / (2 * a);
    cout << "\nx= " << x1;
}
else
{
    cout << "Tenglama ikkita ildizga ega: ";
    x1 = (-b + sqrt(D)) / (2 * a);
    x2 = (-b - sqrt(D)) / (2 * a);
    cout << "\nx1= " << x1;
    cout << "\nx2= " << x2;
}
return 0;
}

```

Дастур бажарилганда, биринчи навбатда тенглама коэффициентлари -  $a$ ,  $b$ ,  $c$  ўзгарувчилар қийматлари киритилади, кейин дискриминант –  $D$  ўзгарувчи қиймати ҳисобланади. Кейин  $D$  қийматининг манфий эканлиги текширилади. Агар шарт ўринли бўлса, яхлит оператор сифатида келувчи ‘{’ ва ‘}’ белгилари орасидаги операторлар бажарилади ва экранга “*Tenglama haqiqiy ildizga ega emas!*” хабари чиқади ва дастур ўз ишини тугатади (“return 0;” операторини бажариш орқали). Дискриминант нолдан кичик бўлмаса, навбатдаги шарт оператори уни нолга тенглигини текширади. Агар шарт ўринли бўлса, кейинги қаторлардаги операторлар блоки бажарилади – экранга “*Tenglama yagona ildizga ega:*” хабари, ҳамда  $x_1$  ўзгарувчи қиймати чоп этилади, акс ҳолда, яъни  $D$  қиймати нолдан катта ҳолати учун **else** калит сўзидан кейинги операторлар блоки бажарилади ва экранга “*Tenglama ikkita ildizga ega:*” хабари, ҳамда  $x_1$  ва  $x_2$  ўзгарувчилар қийматлари чоп этилади. Шу билан шарт операторидан чиқилади ва асосий функциянинг ретурн кўрсатмасини бажариш орқали дастур ўз ишини тугатади.

Ўз навбатида <оператор<sub>1</sub>> ва <оператор<sub>2</sub>> ҳам шартли оператор бўлиши мумкин. Ифодадаги ҳар бир **else** калит сўзи, олдиндаги энг яқин **if** калит

сўзига тегишли ҳисобланади (худди очилувчи ва ёпилувчи қавслардек). Буни инобатга олмаслик мазмунан хатоликларга олиб келиши мумкин.

Масалан:

```
if (x==1)
if (y==1) cout << "x=1 va y=1";
else cout << "x <> 1";
```

Бу мисолда “ $x <> 1$ ” хабари  $x$  қиймати 1 ва  $y$  қиймати 1 бўлмаган ҳолда ҳам чоп этилади. Қуйидаги вариантда ушбу мазмунан хатолик бартараф этилган:

```
if (x==1)
{
    if (y==1) cout << "x=1 va y=1";
}
else cout << "x<>1";
```

Иккинчи мисол тариқасида учта бутун соннинг максимал қийматини топадиган дастур бўлагини келтиришимиз мумкин:

```
int x, y, z, max;
cin >> x >> y >> z;
if (x > y)
    if (y < z) max = z;
    else max = y;
else
    if (x < z) max = z;
    else max = x;
```

## 7.5. ?: шарт амали

Агар текшириладиган шарт нисбатан содда бўлса, шарт амалининг “?:” кўринишини ишлатиш мумкин:

$$\langle \text{шарт ифода} \rangle ? \langle \text{ифода}_1 \rangle : \langle \text{ифода}_2 \rangle;$$

Шарт амали `if` шарт операторига ўхшаш ҳолда ишлайди: агар  $\langle \text{шарт ифода} \rangle$  0 қийматидан фарқли ёки `true` бўлса,  $\langle \text{ифода}_1 \rangle$ , акс ҳолда  $\langle \text{ифода}_2 \rangle$  бажарилади. Одатда ифодалар қийматлари бирорта ўзгарувчига

Ўзлаштирилади. Мисол тариқасида иккита бутун сон максимумини топиш масаласини кўрайлик.

```
if (a >= b)
    max = a;
else
    max = b;
```

Дастурни “?:” оператори ёрдамида қуйидагича ёзиш мумкин:

```
max = (a >= b) ? a : b;
#include <iostream>
using namespace std;
int main()
{
    int a, b, c;
    cout << "a va b sonlar maksimumini topish.";
    cout << "\na - qiymatini kiriting: ";
    cin >> a;
    cout << "\nb - qiymatini kiriting: ";
    cin >> b;
    c = a > b ? a : b;
    cout<< "\nSonlar maksimumi: " << c;
    return 0;
}
```

Дастурдаги шарт оператори қиймат бериш операторининг таркибига кирган бўлиб, **a** ўзгарувчининг қиймати **b** ўзгарувчининг қийматидан катталиги текширилади. Агар шарт рост бўлса, **c** ўзгарувчисига **a** ўзгарувчи қийматини, акс ҳолда **b** ўзгарувчининг қийматини ўзлаштиради ва **c** ўзгарувчисининг қиймати чоп этилади.

**?:** операторининг қиймат қайтариш хоссасидан фойдаланган ҳолда, уни бевосита `cout` кўрсатмасига ёзиш орқали ҳам қўйилган масалани ечиш мумкин:

```
#include <iostream>
using namespace std;
```

```

int main()
{
int a, b;
cout << "a va b sonlar maksimumini topish.";
cout << "\na - qiymatini kiriting: ";
cin >> a;
cout << "\nb - qiymatini kiriting: ";
cin >> b;
cout << "\nSonlar maksimumi: << (a > b) ? a : b;
return 0;
}

```

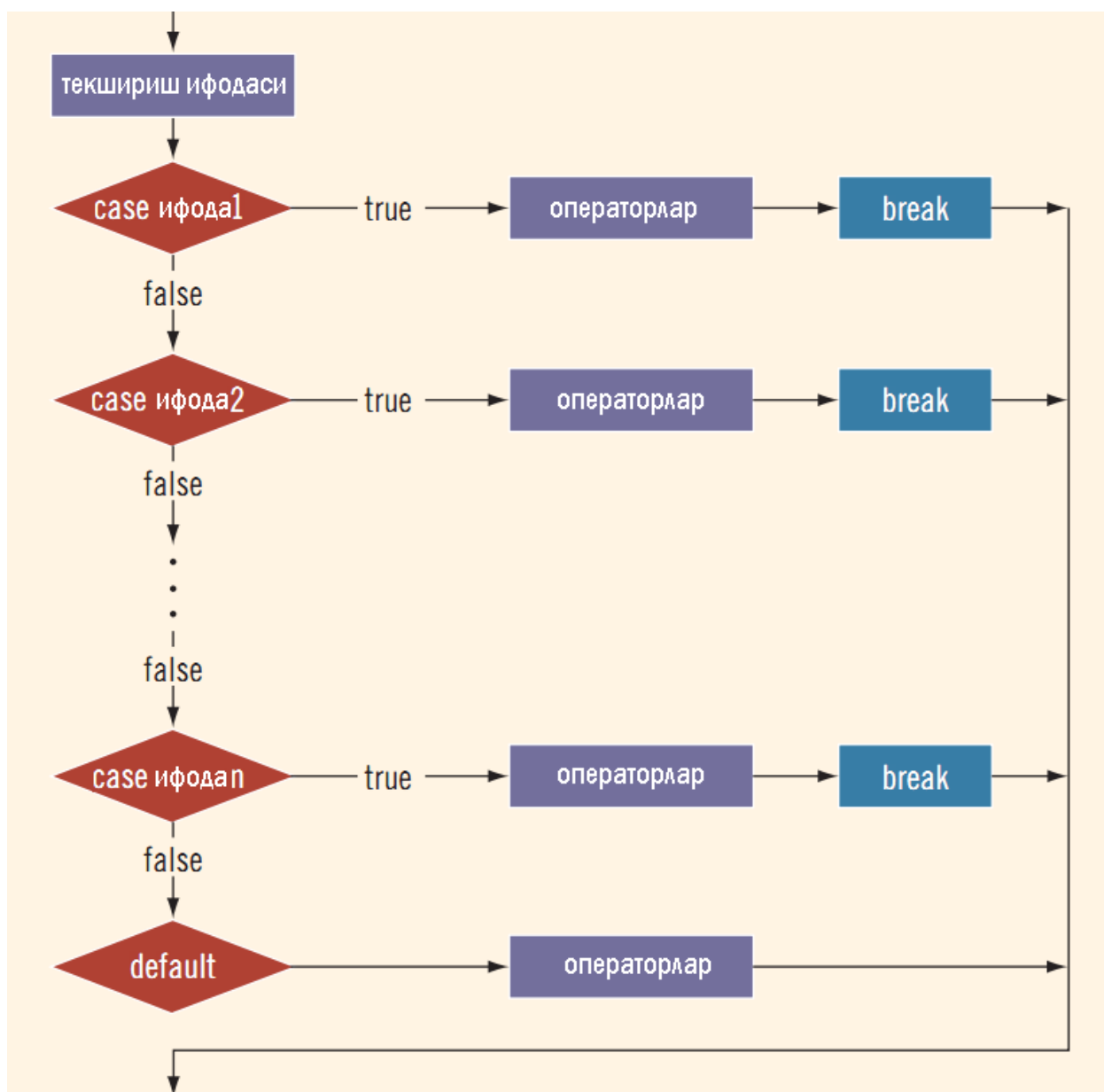
## 7.6. switch operatori

Шарт операторининг яна бир кўриниши switch тармоқланиш operatori бўлиб, унинг синтаксиси қуйидагича:

```

switch (<ifoda>)
{
case <o'zgarmas ifoda_1> :
    <operatorlar guruhi_1>;
    break;
case <o'zgarmas ifoda_2> :
    <operatorlar guruhi_2>;
    break;
.
.
.
case <o'zgarmas ifoda_n> :
    <operatorlar guruhi_n>;
    break;
default :
    <operatorlar guruhi_n+1>;
}

```



Бу оператор қуйидагича амал қилади: биринчи навбатда <ифода> қиймати ҳисобланади, кейин бу қиймат **case** калит сўзи билан ажратилган <ўзгармас ифода<sub>и</sub>> билан солиштирилади. Агар улар устма-уст тушса, шу қатордаги ‘:’ белгисидан бошлаб, токи **break** калит сўзигача бўлган <операторлар гуруҳи<sub>и</sub>> бажарилади ва бошқарув тармоқланувчи оператордан кейин жойлашган операторга ўтади. Агар <ифода> бирорта ҳам <ўзгармас ифода<sub>и</sub>> билан мос келмаса, қурилманинг дефавлт қисмидаги <операторлар гуруҳи<sub>н+1</sub>> бажарилади. Шунини қайд этиш керакки, қурилмада **default** калит сўзи фақат бир марта учраши мумкин.

Намуна учун, **char** туридаги **belgi** ўзгарувчиси орқали текшириш жараёни бажарилаётган қуйидаги намуна қўрилсин:



```

switch (belgi)
{
case 'A':
cout << "Oradagi masofa 10m";
break;
case 'B':
cout << "Oradagi masofa 8m";
break;
case 'C':
cout << "Oradagi masofa 6m";
break;
case 'D':
cout << "Oradagi masofa 3m";
break;
case 'F':
cout << "Oradagi masofa 0m";
break;
default:
cout << "Masofa belgisi noto'g'ri kiritilgan.";
}

```

Ушбу мисолда текшириш учун ўзгарувчи келмоқда. Ўзгарувчининг тури белги бўлгани учун ҳар бир қиймат `case` калит сўзидан кейин келган белги билан солиштирилган.

Мисол учун, кириш оқимидан *“Жараён давом этилсинми?”* сўровига фойдаланувчи томонидан жавоб олинади. Агар ижобий жавоб олинса, экранга *“Жараён давом этади!”* хабари чоп этилади ва дастур ўз ишини тармоқланувчи оператордан кейинги операторларни бажариш билан давом эттиради, акс ҳолда *“Жараён тугади!”* жавоби берилади ва дастур ўз ишини тугатади. Бунда, фойдаланувчининг 'у' ёки 'У' жавоблари жараённи давом эттиришни билдиради, бошқа белгилар эса жараённи тугатишни англатади.

```
#include <iostream>
```

```

using namespace std;
int main()
{
char Javob=' ';
cout << "Jarayon davom etsinmi? ('y','Y'): ";
cin >> Javob;
switch(Javob)
{
    case 'Y':
    case 'y':
        cout << "Jarayon davom etadi!\n";
        break;
    default:
        cout << "Jarayon tugadi!\n";
        return 0;
}
// jarayon
return 0;
}

```

Умуман олганда, тармоқланувчи операторда **бреак** ва **дефаулт** калит сўзларини ишлатиш мажбурий эмас. Лекин бу ҳолатда оператор мазмуни бузилиши мумкин. Масалан, дефаулт қисми бўлмаган ҳолда, агар <ифода> бирорта <ўзгармас ифода<sub>и</sub>> билан устма-уст тушмаса, оператор ҳеч қандай амал бажармасдан бошқарув тармоқланувчи оператордан кейинги операторга ўтади. Агар брек бўлмаса, <ифода> бирорта <ўзгармас ифода<sub>и</sub>> билан устма-уст тушган ҳолда, унга мос келувчи операторлар гуруҳини бажаради ва “*тўхтамасдан*” кейинги қатордаги операторлар гуруҳини ҳам бажаришда давом этади. Масалан, юқоридаги мисолда брек оператори бўлмаса ва жараённи давом эттиришни тасдиқловчи ('Y') жавоб бўлган тақдирда экранга

```
Jarayon davom etadi!
```

## Jarayon tugadi!

хабарлари чиқади ва дастур ўз ишини тугатади (return операторининг бажарилиши натижасида).

Тармоқланувчи оператор санаб ўтилувчи турдаги ўзгармаслар билан биргаликда ишлатилганда самара беради. Қуйидаги дастурда ранглар гаммасини тоифалаш масаласи эчилган.

```
#include <iostream>
using namespace std;
int main()
{
enum Ranglar {Qizil,Tuq_sariq,Sariq,Yashil,
Kuk,Zangori,Binafsha};
Ranglar Rang = 4;
switch (Rang)
{
case Qizil:
case Tuq_sariq:
case Sariq:
    cout << "Issiq gamma tanlandi.\n";
    break;
case Yashil:
case Kuk:
case Zangori:
case Binafsha:
    cout << "Sovuq gamma tanlandi.\n";
    break;
default:
    cout << "Kamalak bunday rangga ega emas.\n";
}
return 0;
```

}

Дастур бажарилишида бошқарув тармоқланувчи операторга кеганда, **Ранг** қиймати **Қизил** ёки **Туқ\_сарик** ёки **Сарик** бўлса, “*Иссиқ гамма танланди*” хабари, агар **Ранг** қиймати **Яшил** ёки **Кук** ёки **Зангори** ёки **Бинафша** бўлса, экранга “*Совуқ гамма танланди*” хабари, агар **Ранг** қиймати санаб ўтилган қийматлардан фарқли бўлса, экранга “*Камалак бундай рангга эга эмас*” хабари чоп этилади ва дастур ўз ишини тугатади.

`switch` операторида эълон операторлари ҳам учраши мумкин. Лекин `switch` оператори бажарилишида “*сакраб ўтиши*” ҳолатлари бўлиши ҳисобига блок ичидаги айрим эълонлар бажарилмаслиги ва бунинг оқибатида дастур ишида хатолик рўй бериши мумкин:

```
//...
int k=0,n=0;
cin >>n;
switch (n)
{
  int i=10;    //xato, bu operator bajarilmaydi
  case 1:
  int j=20;    //agar n=2 bo'lsa, bu e'lon bajarilmaydi
  case 2:
  k+=i+j;     //xato, chunki i,j o'zgaruvchilar noma'lum
}
cout<<k;
//...
```

**Масала.** Қуйида санаб ўтилувчи турлар ва шу турдаги ўзгарувчилар эълон қилинган:

Берилган  $r$  бирликда берилган  $x$  ўзгарувчисининг қиймати метрларда чоп қилинсин.

```
#include <iostream>
using namespace std;
int main()
```

```

    {
        enum Birlik {desimetr, kilometr, metr, millimetr,
santimetr};
        float x,y;
        Birlik r;
        cout << "Uzunlikni kiriting: x=";
        cin>>x;
        cout<<" Uzunlik birliklari\n";
        cout<<" 0- desimetr\n";
        cout<<" 1- kilometr\n";
        cout<<" 2- metr\n";
        cout<<" 3- millimetr\n";
        cout<<" 4- santimetr\n";
        cout<<" Uzunlikni birligini tanlang: r=";
        cin>>r;
        switch(r)
        {
            case desimetr: y=x/10; break;
            case kilometr: y=x*1000; break;
            case metr: y=x; break;
            case millimetr: y=x/1000; break;
            case santimetr: y=x/100; break;
            default:
                cout<<"Uzunlik birligi noto'g'ri kiritildi!";
                return 0;
        }
        cout<<y<<" metr";
        return 0;
    }

```

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим мавзулари

1. Ичма-ич жойлашган шарт операторлари.
2. Танлаш операторининг мураккаб кўринишлари.
3. Танлаш операторини шарт операторлари орқали ифодалаш.
4. Такқослаш амалларининг кўринишлари.

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
cout		экранга чиқариш оқими
cin		экрандан киритиш оқими
if		шарт оператори
ўзгарувчи	variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
Идентификатор	identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизиқ (‘_’) белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги

else		шарт ёлғонлигини аниқловчи оператор
switch		бир нечта константа билан текшириш оператори
case		константалар билан текшириш оператори
default		константага тенг бўлмаганда бажариш оператори

### Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур экрандан киритилган йилни сўз билан ёзиб беради. Лекин дастур киритилган йил фақат мингдан кичик бўлгандагина тўғри натижа беради. Дастур миллионгача бўлган йил киритилганда ҳам уни сўз билан ёзиши керак. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги масалани ишланиш алгоритмини ўрганинг (индивидуал ва кичик гуруҳда).
- Кейсдаги муаммони ҳал қилиш йўлларини аниқланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тилида дастур кодани тақдим этинг (индивидуал ҳолда).

### Назорат саволлари

1. Мантиқий қўшиш оператори нечта ифода орқали ҳисобланади?
2. Мантиқий инкор оператори текширилаётган ифода ёлғон бўлса қандай қиймат қайтаради?
3. **if** оператори нима?
4. C++ тилининг қурилмалари операторларни блок кўринишида ташкил қилишга имкон берадими? Буни тушунтириб беринг.
5. *Блок* – нима?

6. Шарт операторида эълон қилиш операторларини ишлатиш мумкинми?
7. <оператор<sub>1</sub>> ва <оператор<sub>2</sub>> шартли оператор бўлиши мумкинми?
8. Агар текширилаётган шарт нисбатан содда бўлса нима ишлатиш мумкин?
9. switch тармоқланиш оператори нима?
10. break ва default калит сўзлари нимаучун ишлатилади?
11. switch операторида эълон операторлари ҳам учраши мумкинми?
12. switch оператори бажарилишида “сакраб ўтиши” ҳолатлари бўлиши ҳисобига блок ичидаги айрим эълонлар бажарилмаслиги ва бунинг оқибатида дастур ишида хатолик рўй бериши мумкинми?
13. switch оператори нима учун ишлатилади?
14. Санаб ўтилувчи турлар ва шу турдаги ўзгарувчиларга мисол келтиринг.
15. Мантикий операторларга нималар киради?
16. <операнд<sub>1</sub>> <таққослаш амали> < операнд<sub>2</sub>> қуйидаги амал нимани англатади?
17. “&&” “||” “!” операторлари нимани англатади?
18. (x==3) && (y==5) агар x ва y қийматлари ҳар хил бўлса қандай қиймат қайтаради?
19. Мантикий кўпайтириш оператори қандай белги орқали белгиланади?
20. Мантикий қўшиш оператори қандай белги орқали белгиланади?

### Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=4, b=8, y; y=a>b?a:b; cout<<y;
```

- a) 8
- b) 5
- c) 12
- d) 0.5

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=10, b=23;
switch(a) {
case 1: c=a+b; break;
```



```
case 2: c=a*b; break;
default: c=(a+b)*b; break;
}
cout<<c;
```

- a) 759
- b) 33
- c) 230
- d) Хатолик ҳақида хабар беради

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 7, b = 8;
if (a == b++)
    a += a;
else if(a % 2 == 0)
    a += b * b;
else
    a += b; cout<<a;
```

- a) 16
- b) 15
- c) 56
- d) 1

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=1, b=2, s=4;
if (a+b==fabs(a-b)) s=s+a;
if (a-b==fabs(a-b)) s=s-a;
cout << s;
```

- a) 4
- b) 3
- c) 2
- d) 1

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int k=8786, s,b=10,b1=1000;  
if(k%b+k%b1>10) s=(k%b+k/b1)/2;  
else s=10-k%b;  
cout<<s;
```

- a) 7
- b) 8
- c) 6
- d) 3

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a;  
bool t = true, T = false;  
if(t && T)  
a = 100;  
else  
a =200;  
cout <<a<<endl;
```

- a) 200
- b) 50
- c) 150
- d) 300

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a;  
bool t = false, T = false;  
if(t && T)  
a = 100;  
else  
a =200;  
cout <<a<<endl;
```

- a) 200

- b) 100
- c) 400
- d) 500

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a;  
bool t = false, T = true;  
if(t || T)  
a = 500;  
else  
a =800;  
cout <<a<<endl;
```

- a) 500
- b) 1000
- c) 800
- d) 1300

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
#include <iostream.h>  
int main () {  
int a=10, b;  
b=a%10;  
cout<<b;  
return 0;  
}
```

- a) 0
- b) 1
- c) 10
- d) 100

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
#include <iostream.h>
int main () {
int a=10, b;
if (a%2==0) b=a*a;
else
b=-a;
cout<<b;
return 0;
}
```

- a) 100
- b) -10
- c) 10
- d) 0

11. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
#include <iostream.h>
int main () {
int a=10, b=23;
switch(a) {
case 1: c=a+b; break;
case 2: c=a*b; break;
default: c=(a+b)*b; break;
}
cout<<c;
return 0;
}
```

- a) 759
- b) 33
- c) 230
- d) xatolik haqida habar beradi

12. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
#include <iostream.h>
int main() {
int x=1;
x+=5;
cout<<x;
return 0;
}
```

- a) 6
- b) 5
- c) 0
- d) Xatolik haqidagi habar

13. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int main() { float x=1.5, y=-2.6; y=y+x*y; if
(y<x) goto nishon; y=y-x*2; nishon: cout<<y;
return 0; }
```

- a) -6.5
- b) 0.5
- c) -2.82
- d) -6.5

14. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a;
bool t = true, T = true;
if(t && T)
a = 50;
else
a =200;
cout <<a<<endl;
```

- a) 50
- b) 100

- c) 200
- d) 150

15. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a;  
bool t = true, T = false;  
if(t && T)  
a = 100;  
else  
a =200;  
cout <<a<<endl;
```

- a) 200
- b) 50
- c) 150
- d) 300

16. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a;  
bool t = false, T = false;  
if(t && T)  
a = 100;  
else  
a =200;  
cout <<a<<endl;
```

- a) 200
- b) 100
- c) 400
- d) 500

17. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a;  
bool t = false, T = true;
```

```
if(t || T)
a = 500;
else
a =800;
cout <<a<<endl;
```

- a) 500
- b) 1000
- c) 800
- d) 1300

18. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 0, s = 0;
bool t = true, T = true;
for(;t&&T;){
i+=20;
s += i;
if (100 < i)
t = !T;
}
cout <<s<<endl;
```

- a) 420
- b) 530
- c) 100
- d) 890

19. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 0, s = 0;
bool t = false, T = true;
while(t||T){
i+=10;
s += i;
if (i > 100)
T = t;
```

```
}  
cout <<s<<endl;
```

- a) 660
- b) 100
- c) 444
- d) 290

20. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 0, s = 0;  
bool t = false, T = true;  
while(t||T){  
i+=50;  
s += i;  
if (i > 500)  
T = t;  
}  
cout <<s<<endl;
```

- a) 3300
- b) 1200
- c) 2000
- d) 600



## Маъруза 9-10. Такрорлаш операторлари. Бошқарувни узатиш операторлари

### Режа:

1. Такрорланувчи жараёнлар
2. for такрорлаш оператори
3. while такрорлаш оператори
4. do-while такрорлаш оператори
5. break оператори
6. continue оператори
7. goto оператори ва нишонлар

Таянч иборалар: *include, cout, cin, if, for, while, do-while, break, Ўзгарувчи, Идентификатор, else, Чексиз такрорлаш,*

### 9.1. Такрорланувчи жараёнлар

Бешта соннинг ўрта арифметиғи топиш масаласи кўрилсин. Бунинг учун қуйидаги дастур коди қисмидан фойдаланиш мумкин:

```
cin >> num1 >> num2 >> num3 >> num4 >> num5;  
sum = num1 + num2 + num3 + num4 + num5;  
average = sum / 5;
```

Қуйидаги савол туғилиши табиий: сонлар миқдори кўп бўлса нима қилиш керак? Ўзгарувчилар сони кўпайиб кетади. Аммо битта ўзгарувчи билан ҳам ушбу мисолни ечиш мумкин. Бунинг учун қуйидаги дастур коди қисмидан фойдаланиш мумкин:

1. `sum = 0;`
2. `cin >> num;`
3. `sum = sum + num;`

Биринчи ифодада `sum` ўзгарувчисига бошланғич қиймат юкланади. Иккинчи ифодада `num` ўзгарувчисига экран орқали қиймат киритилади.

Учинчи ифодада эса `sum` ўзгарувчисига `num` ўзгарувчисининг қиймати қўшилади.

```
num = 5
```

```
sum = sum + num = 0 + 5 = 5
```

```
num = 3
```

```
sum = sum + num = 5 + 3 = 8 ва хоказо...
```

Агар ўрта арифметиғи топилиши керак бўлган сонлар миқдори кўп бўлса дастур коди кўп бўлиб кетади. Иккинчи ва учинчи ифодани барча сонлар учун такроран ёзиш керак бўлади. Шундай вазиятларда такрорлаш операторларидан фойдаланиш мақсадга мувофиқ.

Такрорлаш оператори “*такрорлаш шарти*” деб номланувчи ифоданинг рост қийматида дастурнинг маълум бир қисмидаги операторларни (такрорлаш танасини) кўп марта такрор равишда бажаради (итератив жараён).

Такрорлаш ўзининг кириш ва чиқиш нуқталарига эга, лекин чиқиш нуқтасининг бўлмаслиғи мумкин. Бу ҳолда такрорлашга *чексиз такрорлаш* дейилади. Чексиз такрорлаш учун такрорлашни давом эттириш шарти доимо рост бўлади.

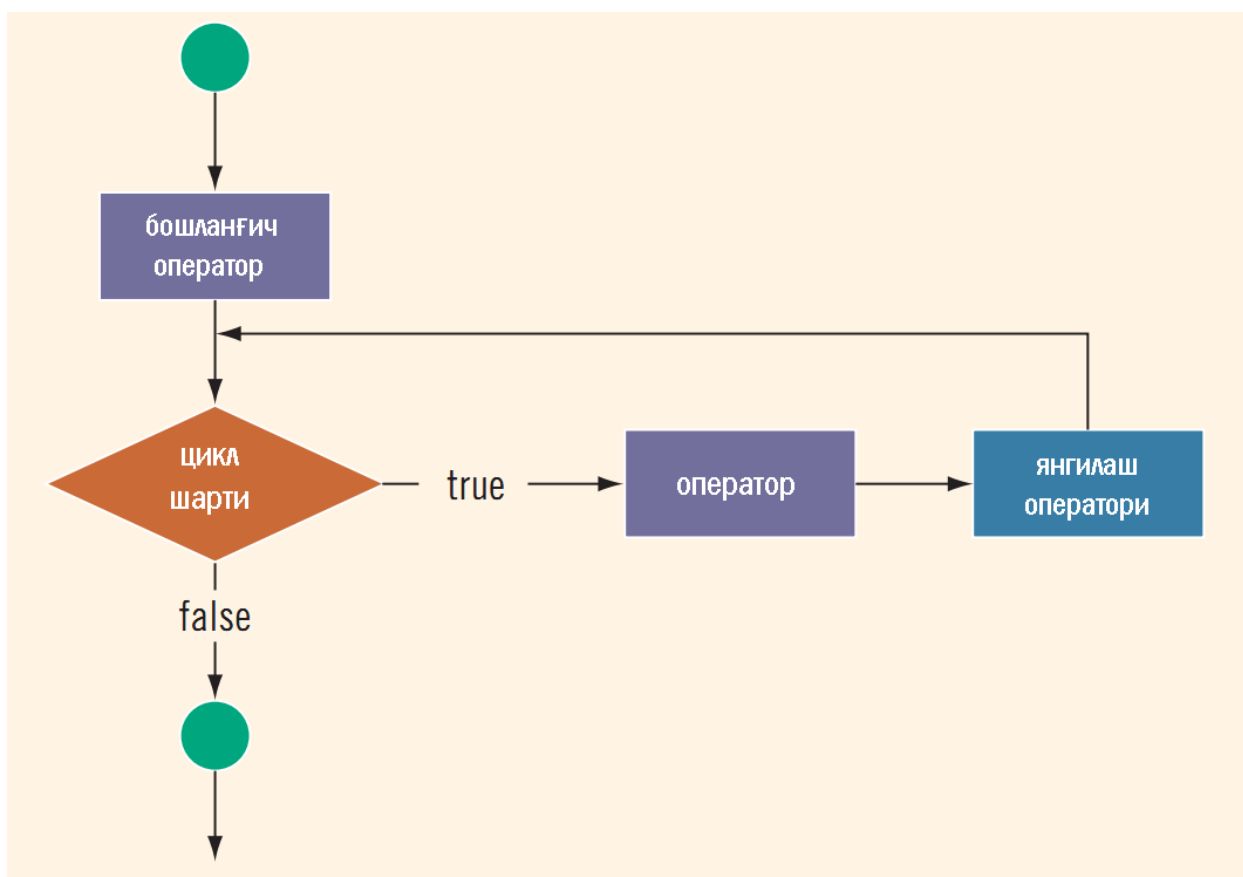
Такрорлаш шартини текшириш такрорлаш танасидаги операторларни бажаришдан олдин текширилиши мумкин (`for`, `while` такрорлашлари) ёки такрорлаш танасидаги операторлари бир марта бажарилгандан кейин текширилиши мумкин (`do-while`).

## 9.2. `for` такрорлаш оператори

`for` такрорлаш операторининг синтаксиси қўйидаги кўринишга эга:

```
for (<ифода1>; <ифода2>; <ифода3>)
```

```
    <оператор ёки блок>;
```



Бу оператор ўз ишини <ифода<sub>1</sub>> ифодасини бажаришдан бошлайди. Кейин такрорлаш қадамлари бошланади. Ҳар бир қадамда <ифода<sub>2</sub>> бажарилади, агар натижа 0 қийматидан фарқли ёки true бўлса, такрорлаш танаси - <оператор ёки блок> бажарилади ва охирида <ифода<sub>3</sub>> бажарилади. Агар <ифода<sub>2</sub>> қиймати 0 (false) бўлса, такрорлаш жараёни тўхтайди ва бошқарув такрорлаш операторидан кейинги операторга ўтади. Шунини қайд қилиш керакки, <ифода<sub>2</sub>> ифодаси вергул билан ажратилган бир нечта ифодалар бирлашмасидан иборат бўлиши мумкин, бу ҳолда охириги ифода қиймати такрорлаш шarti ҳисобланади. Такрорлаш танаси сифатида битта оператор, жумладан бўш оператор бўлиши ёки операторлар блоки келиши мумкин.

Мисол учун 10 дан 20 гача бўлган бутун сонлар йиғиндисини ҳисоблаш масаласини кўрайлик.

```

#include <iostream>
using namespace std;
int main()
{

```

```

int Summa = 0;
for (int i = 10; i <= 20; i++)
    Summa += i;
cout << "Yig'indi=" << Summa;
return 0;
}

```

Дастурдаги такрорлаш оператори ўз ишини,  $i$  такрорлаш параметрига (такрорлаш санагичига) бошланғич қиймат - 10 сонини беришдан бошлайди ва ҳар бир такрорлаш қадамидан (итерациядан) кейин қавс ичидаги учинчи оператор бажарилиши ҳисобига унинг қиймати биттага ошади. Ҳар бир такрорлаш қадамида такрорлаш танасидаги оператор бажарилади, яъни Summa ўзгарувчисига  $i$  қиймати қўшилади. Такрорлаш санагичи  $i$  қиймати 21 бўлганда “ $i \leq 20$ ” такрорлаш шarti false (0 қиймати) бўлади ва такрорлаш тугайди. Натижада бошқарув такрорлаш операторидан кейинги cout операторига ўтади ва экранга йиғинди чоп этилади.

Юқорида келтирилган мисолга қараб такрорлаш операторларининг қавс ичидаги ифодаларига изоҳ бериш мумкин:

<ифода<sub>1</sub>> - такрорлаш санагичи вазифасини бажарувчи ўзгарувчига бошланғич қиймат беришга хизмат қилади ва у такрорлаш жараёни бошида фақат бир марта ҳисобланади. Ифодада ўзгарувчи эълони учраши мумкин ва бу ўзгарувчи такрорлаш оператори танасида амал қилади ва такрорлаш операторидан ташқарида «кўринмайди»;

<ифода<sub>2</sub>> - такрорлашни бажариш ёки йўқлигини аниқлаб берувчи мантикий ифода, агар шарт рост бўлса, такрорлаш давом этади, акс ҳолда йўқ. Агар бу ифода бўш бўлса, шарт доимо рост деб ҳисобланади;

<ифода<sub>3</sub>> - одатда такрорлаш санагичининг қийматини ошириш (камайтириш) учун хизмат қилади ёки унда такрорлаш шартига таъсир қилувчи бошқа амаллар бўлиши мумкин.

C++ тилининг қурилмалари операторларни блок кўринишида ташкил қилишга имкон беради. *Блок* – ‘{’ ва ‘}’ белгилари оралиғига олинган операторлар кетма-кетлиги бўлиб, у компилятор томонидан яхлит бир

оператор деб қабул қилинади. Блок ичида эълон операторлари ҳам бўлиши мумкин ва уларда эълон қилинган ўзгарувчилар фақат шу блок ичида кўринади (амал қилади), блокдан ташқарида кўринмайди. Блокдан кейин ‘;’ белгиси қўйилмаслиги мумкин, лекин блок ичидаги ҳар бир ифода ‘;’ белгиси билан яқунланиши шарт.

```
{
    оператор_1
    оператор_2
    .
    .
    .
    оператор_n
}
```

Такрорлаш операторида ҳам блоклардан фойдаланиш мумкин. Бир нечта операторлар такрорланиши керак бўлганда блоклардан фойдаланиш мумкин. Буни қуйидаги мисолда яққол кўриш мумкин:

```
for (i = 1; i <= 3; i++)
{
    cout << "Hello!" << endl;
    cout << "***" << endl;
}
```

Ушбу мисол натижаси қуйидагича кўринишга эга бўлади:

```
Hello!
***

Hello!
***

Hello!
```

\*\*\*

Агар айнан шу мисолда блок ишлатилмаса дастур қуйидаги кўринишда ишлайди:

```
for (i = 1; i <= 3; i++)  
cout << "Hello!" << endl;  
cout << "***" << endl;
```

Дастур ишлаши натижаси:

Hello!

Hello!

Hello!

\*\*\*

Такрорлаш операторида қавс ичидаги ифодалар бўлмаслиги мумкин, лекин синтаксис ‘;’ бўлмаслигига рухсат бермайди. Шу сабабли, энг содда кўринишдаги такрорлаш оператори қуйидагича бўлади:

```
for ( ; ; )  
    cout << "Cheksiz takrorlash...";
```

Агар такрорлаш жараёнида бир нечта ўзгарувчиларнинг қиймати синхрон равишда ўзгариши керак бўлса, такрорлаш ифодаларида зарур операторларни ‘,’ билан ёзиш орқали бунга эришиш мумкин:

```
for (int i = 10, j = 2 ; i <= 20 ; i++, j = j + 10)  
{  
    s = s + i;  
    p = p + j;  
}
```

Такрорлаш операторининг ҳар бир қадамида  $j$  ва  $i$  ўзгарувчиларнинг қийматлари мос равишда ўзгариб боради.

for операторида такрорлаш танаси бўлмаслиги ҳам мумкин. Масалан, дастур бажарилишини маълум бир муддатга «тўхтатиб» туриш зарур бўлса,

бунга такрорлашни ҳеч қандай қўшимча ишларни бажармасдан амал қилиши орқали эришиш мумкин:

```
#include <iostream>
using namespace std;
int main()
{
    int delay;
    ...
    for (delay = 5000; delay > 0; delay--);
    // bo'sh operator
    ...
    return 0;
}
```

10 дан 20 гача бўлган сонлар йиғиндисини бўш танали такрорлаш оператори орқали ҳисоблаш мумкин:

```
#include <iostream>
using namespace std;
int main()
{
    int Summa = 0;
    for (int i = 10; i <= 20; Summa += i++);
    cout << "Yig'indi=" << Summa;
    return 0;
}
```

Такрорлаш оператори операторлар блоки танаси сифатида ишлатилишини факториални ҳисоблаш мисолида кўрсатиш мумкин:

```
#include <iostream>
```

```

using namespace std;

int main()
{
int a;

unsigned long fact = 1;
cout << "Butun sonni kiriting: ";
cin >> a;
if ((a >= 0) && (a < 33))
{
    for (int i = 1; i <= a; i++)
        fact *= i;
    cout << a << "! = " << fact << '\n';
}
return 0;
}

```

Дастур фойдаланувчи томонидан 0 дан 33 гача ораликдаги сон киритилганда амал қилади, чунки 34! қиймати unsigned long учун ажратилган разрядларга сиғмайди.

Такрорлаш оператори ичма-ич жойлашган бўлиши ҳам мумкин. Бунда ҳар бир ташқарида жойлашган цикл қадами учун ички цикл тўла айланади. Мисол сифатида қаторлар учун қатор сони миқдорига тенг юлдузча белгисини чоп этиш:

```

#include <iostream>

using namespace std;

int main()
{
for (i = 1; i <= 5; i++)
{

```



```

    for (j = 1; j <= i; j++)
        cout << "*" << endl;
    cout << endl;
}
return 0;
}

```

Дастур ишлаши натижаси:

```

*
**
***
****
*****

```

Такрорлаш операторининг ичма-ич жойлашувига мисол сифатида рақамлари бир-бирига ўзаро тенг бўлмаган уч хонали натурал сонларни ўсиш тартибида чоп қилиш масаласини кўриш мумкин:

```

#include <iostream>
using namespace std;
int main()
{
    unsigned char a2,a1,a0;          //uch xonali son
    raqamlari
    for (a2='1';a2<='9';a2++) //sonning 2-raqami
    for (a1='0';a1<='9';a1++) //sonning 1-raqami
    for (a0='0';a0<='9';a0++) //sonning 0-raqami
    // raqamlarni o'zaro teng emasligini tekshirish
    if (a0!=a1 && a1!=a2 && a0!=a2) //o'zaro teng emas
        cout<<a2<<a1<<a0<<' \n';
    return 0;
}

```

}

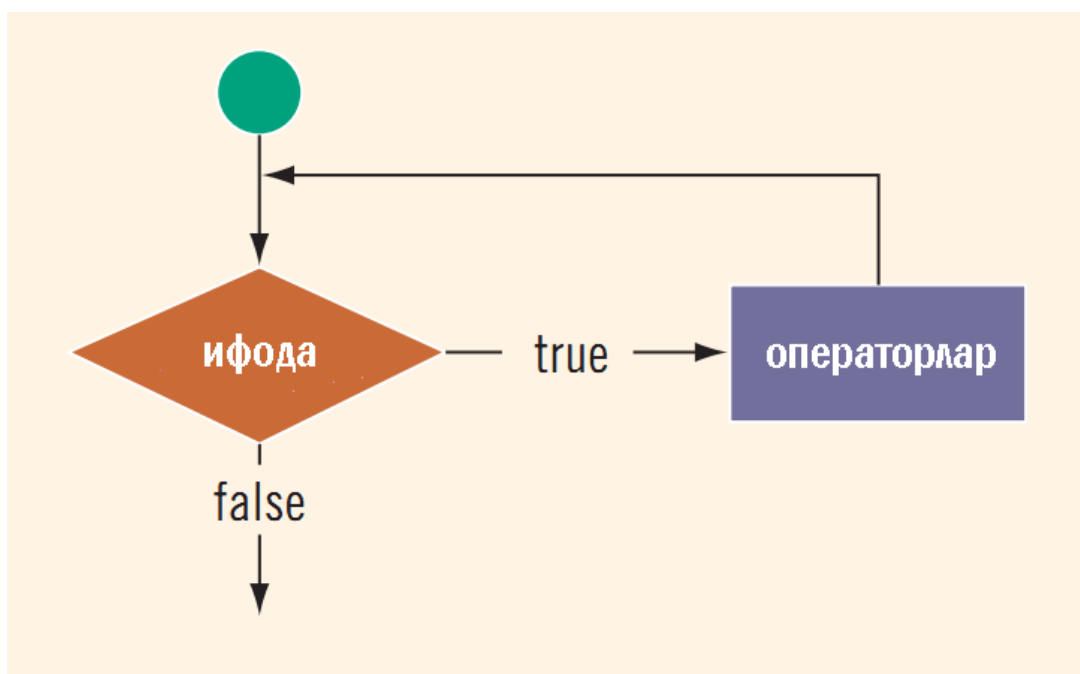
Дастурда уч хонали соннинг ҳар бир рақами такрорлаш операторларининг параметрлари сифатида ҳосил қилинади. Биринчи, ташқи такрорлаш оператори билан 2-хонадаги рақам (**a2** такрорлаш параметри) ҳосил қилинади. Иккинчи, ички такрорлаш операторида (**a1** такрорлаш параметри) сон кўринишининг 1-хонасидаги рақам ва ниҳоят, унга нисбатан ички бўлган **a0** параметрли такрорлаш операторида 0-хонадаги рақамлар ҳосил қилинади. Ҳар бир ташқи такрорлашнинг бир қадамига ички такрорлаш операторининг тўлиқ бажарилиши тўғри келиши ҳисобига барча уч хонали сонлар кўриниши ҳосил қилинади.

### 9.3. while такрорлаш оператори

while такрорлаш оператори, оператор ёки блокни такрорлаш шarti ёлғон (false ёки 0) бўлгунча такрор бажаради. У қуйидаги синтаксисга эга:

```
while (<ifoda>)
```

```
<operator yoki blok>;
```



Агар <ифода> рост қийматли ўзгармас ифода бўлса, такрорлаш чексиз бўлади. Худди шундай, <ифода> такрорлаш бошланишида рост бўлиб, унинг қийматига такрорлаш танасидаги ҳисоблаш таъсир этмаса, яъни унинг қиймати ўзгармаса, такрорлаш чексиз бўлади.

while такрорлаш шартини олдиндан текширувчи такрорлаш оператори ҳисобланади. Агар такрорлаш бошида <ифода> ёлғон бўлса, while оператори таркибидаги <оператор ёки блок> қисми бажарилмасдан чеклаб ўтилади.

```
i = 0;
while (i <= 20)
{
    cout << i << " ";
    i = i + 5;
}
cout << endl;
```

Дастур қисми ишлаши натижаси:

```
0 5 10 15 20
```

Айрим ҳолларда <ифода> қиймат бериш оператори кўринишида келиши мумкин. Бунда қиймат бериш амали бажарилади ва натижа 0 билан солиштирилади. Натижа нолдан фарқли бўлса, такрорлаш давом эттирилади.

Агар рост ифоданинг қиймати нолдан фарқли ўзгармас бўлса, чексиз такрорлаш рўй беради. Масалан:

```
while (1); // чексиз такрорлаш
```

Худди for операторидек, ‘,’ ёрдамида <ифода> да бир нечта амаллар синхрон равишда бажариш мумкин. Масалан, сон ва унинг квадратларини чоп қиладиган дастурда ушбу ҳолат кўрсатилган:

```
#include <iostream>
using namespace std;
int main()
{
    int n,n2;
    cout << "Sonni kiriting(1..10): ";
    cin >> n;
```

```

n++;
while (n--, n2 = n * n , n>0)
cout << " n=" << n << " n^2 = " << n2 << endl;
return 0;
}

```

Дастурдаги такрорлаш оператори бажарилишида  $n$  сони 1 гача камайиб боради. Ҳар бир қадамда  $n$  ва унинг квадрати чоп қилинади. Шунга эътибор бериш керакки, шарт ифодасида операторларни ёзилиш кетма-кетлигининг аҳамияти бор, чунки энг охириги оператор такрорлаш шарти сифатида қаралади ва  $n$  қиймати 0 бўлганда такрорлаш тугайди.

Кейинги дастурда берилган ўнлик соннинг иккилик кўринишини чоп қилиш масаласини ечишда `while` операторини қўллаш кўрсатилган.

```

#include <iostream>
using namespace std;
int main()
{
int sanagich = 4;
short son10, jarayon = 1;
while (jarayon) // cheksiz takrorlash
{
cout << "0'nlik sonni kiriting(0..15) ";
cin >> son10;
cout << '\n' << son10
<< "Sonining ikkilik ko'rinishi: ";
while (sanagich)
{
if (son10 & 8) //son10 & 00001000
cout << '1';
}
}
}

```

```

else cout << '0';

son10 = son10 << 1;

//razryadlarni chapga surish
sanagich--;

}

cout << '\n' ;

cout << "Jarayonni to'xtasin(0), davom etsin(1): ";

cin >> jarayon;

sanagich = 4;

}

return 0;

}

```

Дастурда ичма-ич жойлашган такрорлаш операторлари ишлатилган. Биринчиси, соннинг иккилик кўринишини чоп қилиш жараёнини давом эттириш шарти бўйича амал қилади. Ички жойлашган иккинчи такрорлаш операторидаги амаллар - ҳар қандай, 0 дан 15 гача бўлган сонлар тўртта разрядли иккилик сон кўринишида бўлишига асосланган. Унда киритилган соннинг ички, иккилик кўринишида учинчи разрядида 0 ёки 1 турганлиги аниқланади (“son10 & 8”). Шарт натижаси натижа 1 (рост) бўлса, экранга ‘1’, акс ҳолда ‘0’ белгиси чоп этилади. Кейинги кадамда сон разрядлари чапга биттага сурилади ва яна учинчи разряддаги рақам чоп этилади. Такрорлаш санагич қиймати 0 бўлгунча яъни тўрт марта бажарилади ва бошқарув ички такрорлаш операторидан чиқади.

while такрорлаш оператори ёрдамида самарали дастур коди ёзишга яна бир мисол бу - иккита натурал сонларнинг энг катта умумий бўлувчисини (ЕКУБ) Эвклид алгоритми билан топиш масаласини келтиришимиз мумкин:

```

#include <iostream>

using namespace std;

int main()

```

```

{
int a,b;
cout << "A va B natural sonlar EKUBini topish.\n";
cout << "A va B natural sonlarni kiriting: ";
cin >> a >> b;
while (a != b)
    a > b ? a -= b : b -= a;
cout << "Bu sonlar EKUBi = " << a;
return 0;
}

```

Бутун турдаги  $a$  ва  $b$  қийматлари оқимдан ўқилгандан кейин токи уларнинг қийматлари ўзаро тенг бўлмагунча такрорлаш жараёни рўй беради. Такрорлашнинг ҳар бир қадамида  $a$  ва  $b$  сонларнинг каттасидан кичиги айрилади. Такрорлашдан кейинги кўрсатма воситасида  $a$  ўзгарувчисининг қиймати натижа сифатида чоп этилади.

#### 9.4. do-while такрорлаш оператори

do-while такрорлаш оператори while операторидан фарқли равишда олдин оператор ёки блокни бажаради, кейин такрорлаш шартини текширади. Бу қурилма такрорлаш танасини камида бир марта бажарилишини таъминлайди. do-while такрорлаш оператори қуйидаги синтаксисга эга:

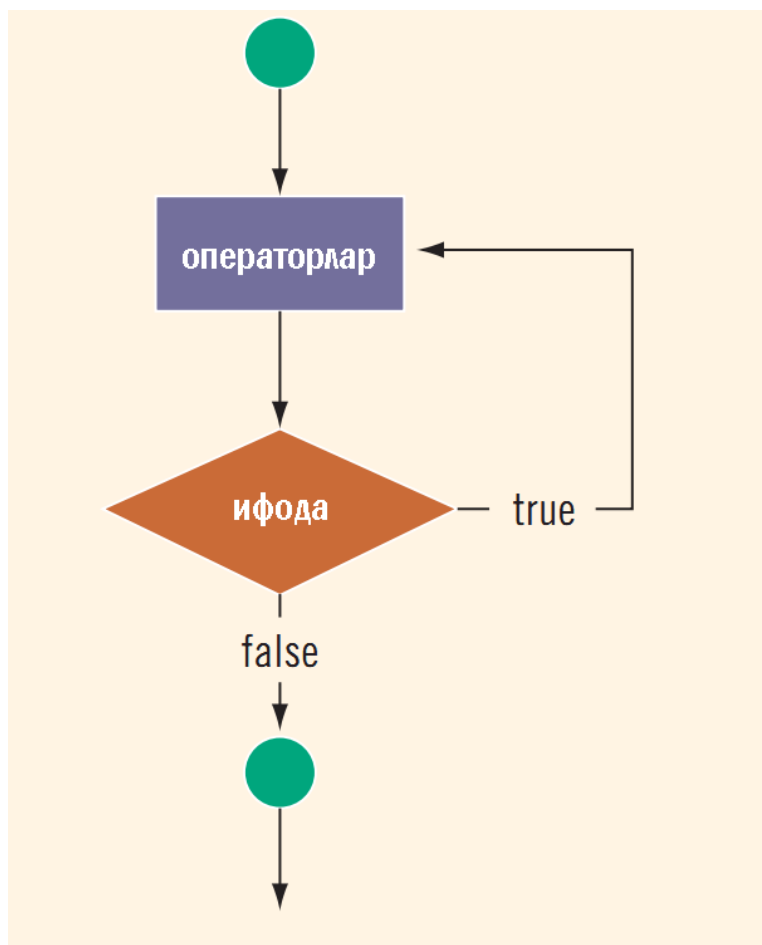
```

do
    <оператор ёки блок>;
while (<ифода>);

```

Бундай такрорлаш операторининг кенг қўлланиладиган ҳолатлари - бу такрорлашни бошламасдан туриб, такрорлаш шартини текширишнинг иложи бўлмаган ҳолатлар ҳисобланади. Масалан, бирорта жараённи давом эттириш ёки тўхтатиш ҳақидаги сўровга жавоб олиш ва уни текшириш зарур бўлсин. Кўриниб турибдики, жараённи бошламасдан олдин бу сўровни беришнинг

маъноси йўқ. Ҳеч бўлмаганда такрорлаш жараёнининг битта қадами амалга оширилган бўлиши керак.



```
#include <iostream>
using namespace std;
int main()
{
    char javob;
    do
    {
        cout << "dastur tanasi\n";
        cout << "Jarayonni to'xtatish (N): ";
        cin >> javob;
    }
    while (javob != 'N');
```

```
return 0;
}
```

Дастур токи “Jarayonni to’xtatish (N):” сўровига 'N' белгиси (жавоби) киритилмагунча давом этади.

Бу оператор ҳам чексиз такрорланиши мумкин:

```
do
{
    cout << "cheksiz sikl tanasi ";
}
while(1);
```

do-while такрорлаш оператори ҳам бошқа такрорлаш операторлари ичма ич жойлашиб келиши мумкин. **Масала:** Ҳар қандай 7 катта бутун сондаги пул миқдорини 3 ва 5 сўмликларда бериш мумкинлиги исботлансин. Қўйилган масала  $p=3n+5m$  тенгламаси қаноатлантирувчи  $m$ ,  $n$  сонлар жуфтликларини топиш масаласидир ( $p$  – пул миқдори). Бу шартнинг бажарилишини  $m$  ва  $n$  ўзгарувчиларининг мумкин бўлган қийматларининг барча комбинацияларида текшириш зарур бўлади.

```
#include <iostream>
using namespace std;
int main()
{
    unsigned int Pul;
        unsigned n3, m5;
    bool xato = false;
    do
    {
        if (xato) cout<<"Pul qiymati 7 dan kichik!";
        xato = true;
```



```

        cout << "\nPul qiymatini kiriting (>7): ";
        cin >> Pul;
    }
    while (Pul <= 7);
    n3=0;
    do
    {
        m5=0;
        do
        {
            if (3 * n3 + 5 * m5 == Pul)
                cout << n3 << " ta 3 so'mlik + "
                    << m5 << " ta 5 so'mlik\n";
            m5++;
        }
        while(3 * n3 + 5 * m5 <= Pul);
        n3++;
    }
    while(3 * n3 <= Pul);
    return 0;
}

```

Дастур пул қийматини киритишни сўрайди (Pul ўзгарувчисига). Агар пул қиймати 7 сонидан кичик бўлса, бу ҳақда хабар берилади ва такрор равишда қиймат киритиш талаб қилинади. Пул қиймати 7 дан катта бўлганда, 3 ва 5 сўмликларнинг мумкин бўлган тўла комбинатсиясини амалга ошириш учун ичма-ич такрорлашлар амалга оширилади. Ташқи такрорлаш n3 (3 сўмликлар миқдори) бўйича, ички такрорлаш эса m5 (5 сўмликлар миқдори) бўйича, токи бу миқдордаги пуллар қиймати Пул қийматидан ошиб

кетмагунча давом этади. Ички такрорлашда  $m5$  ўзгарувчисининг ҳар бир қийматида « $3 * n3 + 5 * m5 == Pul$ » шарти текширилади, агар у ўринли бўлса, ечим варианты сифатида  $n3$  ва  $m5$  ўзгарувчилар қийматлари чоп этилади.

Пул қиймати 30 сўм киритилганда ( $Pul=30$ ), экранга

0 ta 3 so'mlik + 6 ta 5 so'mlik

5 ta 3 so'mlik + 6 ta 5 so'mlik

10 ta 3 so'mlik + 0 ta 5 so'mlik

ечим вариантлари чоп этилади.

## 9.5. break оператори

Такрорлаш операторларининг бажарилишида шундай ҳолатлар юзага келиши мумкинки, унда қайсидир қадамда, такрорлашни якунига этказмасдан такрорлашдан чиқиш зарурати бўлиши мумкин. Бошқача айтганда, такрорлашни «узиш» керак бўлиши мумкин. Бунда break операторидан фойдаланилади. break операторини такрорлаш оператори танасининг ихтиёрий (зарур) жойларига қўйиш орқали шу жойлардан такрорлашдан чиқишни амалга ошириш мумкин. Эътибор берадиган бўлсак, switch-case операторининг туб моҳиятига ҳам break операторини қўллаш орқали эришилган.

Ичма - ич жойлашган такрорлаш ва switch операторларида break оператори фақат ўзи жойлашган блокдан чиқиш имкониятини беради.

Қуйидаги дастурда иккита ичма-ич жойлашган такрорлаш операторидан фойдаланган ҳолда фойдаланувчи томонидан киритилган қандайдир сонни 3 ва 7 сонларига нисбатан қандай ораликқа тушиши аниқланади. Ташқи такрорлашда “Сон киритинг (0- тўхташ):\_” сўрови берилади ва жавоб жавоб\_сон ўзгарувчисига ўқилади. Агар сон нолдан фарқли бўлса, ички такрорлаш операторида бу соннинг қандайдир ораликқа тушиши аниқланиб, шу ҳақида хабар берилади ва ички такрорлаш операторидан чиқилади. Ташқи такрорлашдаги сўровга жавоб тариқасида 0 киритилса, дастур ўз ишини тугатади.

```
#include <iostream>
```

```

using namespace std;

int main()
{
int javob_son=0;
do
{
while (javob_son)
{
if(javob_son < 3)
{
cout << "3 kichik!";
break;
}
if(3 <= javob_son && javob_son <= 7)
{
cout << "3 va 7 oraligida !";
break;
}
if(javob_son > 7)
{
cout << "7 dan katta !";
break;
}
}
cout << "\nSon kiriting (0-to'xtash):_";
cin >> javob_son;
}

```

```

while(javob_son != 0);
return 0;
}

```

Амалиётда break операторидан чексиз такрорлашдан чиқишда фойдаланилади.

```

for ( ; ; )
{
    // 1- shart
    if (...)
    {
        ...
        break;
    }
    // 2- shart
    if (...)
    {
        ...
        break;
    }
    ...
}

```

Бу мисолда чексиз фор такрорлашидан 1 ёки 2 - шарт бажарилганда чиқилади.

**Масала.** Ишорасиз бутун сонлар кетма-кетлиги 0 қиймати билан тугайди, 0 кетма-кетлик ҳади ҳисобланмайди. Кетма-кетликни камаймайдиган ҳолда тартибланган ёки йўқлиги аниқлансин.

```

#include <iostream>

using namespace std;

```

```

int main()
{
    unsigned int Ai_1=0,Ai;
    cout<<"Sonlar ketma-ketligini kiriting"
    cout<<("0-tugash alomati):\n ";
    cin>>Ai;
    while(Ai)
    {
        Ai_1=Ai;
        cin>>Ai;
        if (Ai_1>Ai) break;
    }
    if(Ai_1)
    {
        cout<<"Ketma-ketlik tartiblangan";
        if(!Ai)cout<<" emas!";
        else cout<<"!";
    }
    else cout<<"Ketma-ketlik bo'sh!";
    return 0;
}

```

Дастур ишга тушганда, бошда кетма-кетликнинг биринчи ҳади алоҳида ўқиб олинади ( $A_i$  ўзгарувчисига). Кейин  $A_i$  қиймати нолга тенг бўлмагунча такрорлаш оператори амал қилади. Такрорлаш танасида  $A_i$  қиймати олдинги қиймат сифатида  $A_{i-1}$  ўзгарувчисида эслаб қолинади ва навбатдаги ҳад  $A_i$  ўзгарувчисига ўқилади. Агар олдинги ҳад навбатдаги ҳаддан катта бўлса, break оператори ёрдамида такрорлаш жараёни узилади ва бошқарув такрорлашдан кейинги шарт операторига ўтади. Бу эрдаги шарт операторлари

мазмуни қуйидагича: агар  $A_{i-1}$  нолдан фарқли бўлса, кетма-кетликнинг камида битта ҳади киритилган бўлади (кетма-кетлик мавжуд) ва охириги киритилган ҳад текширилади. Ўз навбатида агар  $A_i$  нолдан фарқли бўлса, бу ҳолат ҳадлар ўртасида камаймаслик шарти бажарилмаган-лиги сабабли ҳадларни киритиш жараёни узилганини билдиради ва бу ҳақда хабар чоп этилади. Акс ҳолда кетма-кетликни камаймайдиган ҳолда тартибланган бўлади.

## 9.6. continue оператори

continue оператори худди break операторидек такрорлаш оператори танасини бажаришни тўхтатади, лекин такрорлашдан чиқиб кетмасдан кейинги қадамига «сакраб» ўтишини тайинлайди.

continue операторини қўлланишига мисол тариқасида 2 ва 50 сонлар оралиғидаги тўб сонларни топадиган дастур матнини келтирамиз.

```
#include <iostream>
using namespace std;
int main()
{
    bool bulinadi = false;
    for (int i = 2; i < 50; i++)
    {
        for (int j = 2; j < i / 2; j++)
        {
            if (i % j) continue;
            bulinadi = true;
            break;
        }
        // break bajarilganda boshqaruv o'tadigan joy
        if (!bulinadi) cout << i << " ";
    }
}
```

```

        bulinadi = false;
    }
    return 0;
}

```

Келтирилган дастурда қўйилган масала ичма-ич жойлашган иккита такрорлаш операторлари ёрдамида эчилган. Биринчи такрорлаш оператори 2 дан 50 гача сонларни ҳосил қилишга хизмат қилади. Ички такрорлаш эса ҳар бир ҳосил қилинаётган сонни 2 сонидан токи шу соннинг ярмигача бўлган сонларга бўлиб, қолдиғини текширади, агар қолдиқ 0 сонидан фарқли бўлса, навбатдаги сонга бўлиш давом этади, акс ҳолда булинади ўзгарувчисига true қиймат бериб, ички такрорлаш узилади (сон ўзининг ярмигача бўлган қандайдир сонга бўлинар экан, демак у туб эмас ва кейинги сонларга бўлиб текширишга ҳожат йўқ). Ички j бўйича такрорлашдан чиққандан кейин булинади қиймати false бўлса (!bulinadi), i сони туб бўлади ва у чоп қилинади.

### 9.7. goto оператори ва нишонлар

*Нишон* - бу давомида иккита нуқта (':') қўйилган идентификатор. Нишон билан қандайдир оператор белгиланади ва кейинчалик, дастурнинг бошқа бир қисмидан унга шартсиз ўтиш амалга оширилади. Нишон билан ҳар қандай оператор белгиланиши мумкин, шу жумладан эълон оператори ва бўш оператори ҳам. Нишон фақат функциялар ичида амал қилади.

Нишонга шартсиз ўтиш goto оператори ёрдамида бажарилади. goto оператори орқали фақат унинг ўзи жойлашган функция ичидаги операторларга ўтиш мумкин. goto операторининг синтаксиси қуйидагича:

```
goto <нишон>;
```

Айрим ҳолларда, goto операторининг «сакраб ўтиши» ҳисобига хатоликлар юзага келиши мумкин. Масалан,

```

int i=0, j=0;

i++;

if (i == 1) goto m;

```

```
j += 5;  
m: j += i;
```

Шартсиз ўтиш оператори дастурни тузишдаги кучли ва шу билан биргаликда хавфли воситалардан бири ҳисобланади. Кучлилиги шундаки, унинг ёрдамида алгоритмнинг «боши берк» жойларидан чиқиб кетиш мумкин. Иккинчи томондан, блокларнинг ичига ўтиш, масалан, такрорлаш операторларини ичига «сакраб» кириш кутилмаган ҳолатларни юзага келтириши мумкин.

Гарчи, нишон ёрдамида дастурнинг ихтиёрий жойига ўтиш мумкин бўлса ҳам, бошланғич қиймат бериш эълонларидан сакраб ўтиш ман этилади, лекин блоклардан сакраб ўтиш мумкин.

Қуйидаги дастурда иккита натурал соннинг энг катта умумий бўлувчисини (ЕКУБ) топиш масаласидаги такрорлаш жараёнини нишон ва goto оператори воситасида амалга ошириш кўрсатилган:

```
#include <iostream>  
using namespace std;  
int main()  
{  
    int a,b;  
    cout << "A va B natural sonlar EKUBini topish.\n";  
    cout << "A va B natural sonlarni kiriting: ";  
    cin >> a >> b;  
    nishon:  
    if(a == b)  
    {  
        cout << "Bu sonlar EKUBi: " << a;  
        return 0;  
    }  
    a > b ? a -= b : b -= a;
```



```
goto nishon;
}
```

Дастурдаги нишон билан белгиланган операторда а ва b сонларни тенглиги текширилади. Агар улар тенг бўлса, ихтиёрий биттаси, масалан а сони ЭКУБ бўлади ва функциядан чиқилади. Акс ҳолда, бу сонларнинг каттасидан кичиги айрилади ва goto орқали уларнинг тенглиги текширилади. Такрорлаш жараёни а ва b сонлар ўзаро тенг бўлгунча давом этади.

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустакил таълим мавзулари:

1. Ичма-ич жойлашган такрорланувчи жараёнлар.
2. Такрорлаш операторларининг мураккаб кўринишлари.
3. Такрорлаш ва шарт операторларидан биргаликда фойдаланиш.
4. goto оператори ёрдамида такрорланувчи жараёнларни ҳосил қилиш.

### Глосарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади.
cout		экранга чиқариш оқими.
cin		экрандан киритиш оқими.
if		шарт оператори

for		такрорлаш қадами билан бериладиган такрорлаш оператори
while		шарти олдин текшириладиган такрорлаш оператори
do-while		шарти кейин текшириладиган такрорлаш оператори
break		такрорлашни тўхтатиш оператори
continue		такрорлаш кейинги қадамига ўтказиш оператори
Ўзгарувчи	Variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги.
Идентификатор	Identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизик ('_') белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги.
else		шарт ёлғонлигини аниқловчи оператор
Чексиз такрорлаш	Endless loop	такрорлашни тўхтатиш шартининг мавжуд эмаслиги

## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур биринчи  $n$  та иккиннинг даражаларини индексларини чоп этади. Лекин дастур ўз ишини тўхтатмасдан давом этмоқда. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

### Назорот саволлари:

1. Бешта соннинг ўрта арифметиги қандай топилади?
2. Чексиз такрорлаш учун такрорлашни давом эттириш шарти?
3. Такрорлаш операторида ҳам блоклардан фойдаланиш мумкинми?
4. C++ тилининг қурилмалари операторларни блок кўринишида ташкил қилишга имкон берадими?
5. Агар <ифода> рост қийматли ўзгармас ифода бўлса, такрорлаш қандай булади?
6. Такрорлаш операторларининг бажарилишида қандай ҳолатлар юзага келиши мумкин?
7. Такрорлаш оператори ичма-ич жойлашган бўлиши мумкинми?
8. do-while такрорлаш оператори қандай вазифани бажаради?
9. while такрорлаш шартини олдиндан текширувчи такрорлаш оператори ҳисобланадими?
10. else оператори қандай вазифани бажаради?
11. continue оператори қандай вазифани бажаради?
12. Break оператори қандай вазифани бажаради?
13. While оператори қандай вазифани бажаради?
14. For оператори қандай вазифани бажаради?
15. If оператори қандай вазифани бажаради?
16. Cin оператори қандай вазифани бажаради?
17. Cout оператори қандай вазифани бажаради?
18. Include оператори қандай вазифани бажаради?

19. Айрим ҳолларда, гого операторининг «сақраб ўтиши» ҳисобига хатоликлар юзага келиши мумкунми?
20. Такрорлаш операторида кавс ичидаги ифодалар бўлмаслиги мумкин, лекин синтаксис ‘;’ бўлмаслигига рухсат бериладими?

### Тест саволлари:

1. for (<ifoda >1; <ifoda>2;<ifoda>3) учун нотўғри тавсифни кўрсатинг?
- a) <ifoda>2 – такрорлаш санагичи вазифасини бажарувчи ўзгарувчига бошланғич қиймат беришга хизмат қилади
  - b) <ifoda>2 – такрорлашни бажариш ёки йўқлигини аниқлаб берувчи мантикий ифода, агар шарт рост бўлса такрорлаш давом этади
  - c) <ifoda>3 – одатда такрорлаш санагичи қийматини ошириш (камайтириш) учун хизмат қилади ёки бу ерда такрорлаш шартига таъсир қилувчи бошқа амаллар бўлиши мумкин.
  - d) <ifoda>1 - такрорлаш санагичи вазифасини бажарувчи ўзгарувчига бошланғич қиймат беришга хизмат қилади
2. int n=10; while(n-=1, n2=n\*n, n>0); Дастур кодида такрорлашни тўхташига таъсир қилувчи шартини топинг?
- a) n>0
  - b) n-=1
  - c) n2=n\*n
  - d) n-=1, n2=n\*n
3. Чексиз такрорлашдан қайси оператор ёрдамида чиқиб кетиш мумкин?
- a) break;
  - b) continue;
  - c) return;
  - d) switch
4. C++ тилида for такрорлаш операторининг синтаксиси тўғри берилган каторни топинг?
- a) for(<ifoda1>;<ifoda2>; <ifoda3>)<operator yoki blok>
  - b) for(<ifoda>)operator yoki blok>
  - c) for(<ifoda1>,<ifoda2>, <ifoda3>)<operator yoki blok>
  - d) for(<ifoda1>;<ifoda2>; <ifoda3>)<operator

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
#include <iostream.h>
int main () {
int s=0, i;
for (i=1; i<10; i++)
s+=i;
cout<<s;
return 0; }
```

- a) 45
- b) 55
- c) 10
- d) 1

6. C++ тилида чексиз такрорлаш while оператори ёрдамида қандай тасвирланади?

- a) while(1)
- b) while(0)
- c) while()
- d) while(false)

7. C++ тилида олдин операторни ёки блокни бажариб, кейин шартни текширувчи оператор қайси?

- a) do-while
- b) while
- c) for
- d) switch

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади? int i,s=0;

```
for (i=1; i<10; i++) {
s+=i;
i++;
}
cout<<s;
```

a) 25

```
int s = 1;
for(int i = 0; i <= 6; i += 2)
    s+=i;
```

b) 45

c) 1

d) 10

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади? I

```
int i, s=0;
for(i = 0; i < 6; i++)
{
    s *= i;
}
cout << s ;
```

a) 0

b) 120

c) 720

d) 24

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i, s=1;
for(i = 0; i < 6; i++)
{
    s *= i;
}
cout << s ;
```

a) 0

b) 720

c) 24

d) 120

11. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
cout << s ;
```

- a) 13
- b) 12
- c) 22
- d) 21

12. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int n=7;
```

```
bool k;
```

```
k=n%2;
```

```
cout << k;
```

- a) 1
- b) True
- c) False
- d) 0

13. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int n=123, a;
```

```
a=n%100;
```

```
cout << a;
```

- a) 23
- b) 12
- c) 3
- d) 0.3

14. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int n = 20, s = 0;
```

```
for (int i = 1; i <= 20; i++)
```

```
    if (n % i == 0)
```

```
        s += i;
```

```
    cout << s << endl;
```

- a)42
- b)100
- c)55
- d)45

15. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int s= 0;
for (int i = 1; i <= 10; i++)
    switch ( i % 5 )
{
    case 0: s += i;
break;
default : C++;
}
cout << s << endl;
```

- a)23
- b)43
- c)24
- d)21

16. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int s = 0, i = 0, n=10;
    dastur_uz : i++;
s += i;
if (i != n)
    goto dastur_uz;
else
    s *= 2;
cout << s << endl;
```

- a)110
- b)120
- c)125
- d)250



17. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int n=10, s = 0, i = 0;
while (1 > 0)
{ i++;          s += i;
if ( i >= n ) break;}
cout << s << endl;
```

- a)55
- b)45
- c)0
- d)Cheksiz takrorlanib qoladi

18. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=24, b=18;
while (a != b)
    if (a > b) a -= b;
    else b -= a;
cout << a << endl;
```

- a)6
- b)2
- c)12
- d)1

19. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 0, j = 0;
bool b = true;
while(b){
i += 2; if (i == 10)
b = false; j = j + i;}
cout << j << endl;
```

- a)30
- b)40
- c)50
- d)20

20. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int s = 0, a = 100, b = 120;
bool t = true;
do{
if(a > b){ t = false;          s = a + b;
} else      b = a/2;
} while(t);
cout <<s << endl;
```

- a)150
- b)200
- c)250
- d)100

## Маъруза 11-12. Статик массивлар

### Режа:

1. Массивлар ҳақида тушунча.
2. Массив элементига мурожаат қилиш
3. Кўп ўлчамли массивлар
4. Бир ўлчамли массивларни қайта ишлаш
5. Бир ўлчамли массивларни инициализация қилиш
6. Кўп ўлчамли массивларни инициализациялаш

Таянч иборалар: *include, cout, cin, for, while, do-while, char, iostream, индекс, массив, вектор, матрица.*

### 11.1. Массивлар ҳақида тушунча

Хотирада кетма-кет (регуляр) жойлашган бир хил турдаги қийматларга массив дейилади. Одатда массивларга зарурат, катта ҳажмдаги, лекин чекланган миқдордаги ва тартибланган қийматларни қайта ишлаш билан боғлиқ масалаларни эчишда юзага келади. Фараз қилайлик, талабалар гуруҳининг рейтинг баллари билан ишлаш масаласи кўйилган. Унда гуруҳнинг ўртача рейтингини аниқлаш, рейтингларни камайиши бўйича тартиблаш, конкрет талабанинг рейтингини ҳақида маълумот бериш ва бошқа масала остиларини эчиш зарур бўлсин. Қайд этилган масалаларни эчиш учун берилганларнинг (рейтингларнинг) тартиб-ланган кетма-кетлиги зарур бўлади. Бу ерда тартибланганлик маъноси шундаки, кетма-кетликнинг ҳар бир қиймати ўз ўрнига эга бўлади (биринчи талабанинг рейтингини массивда биринчи ўринда, иккинчи талабаники - иккинчи ўринда ва ҳақоза). Берилганлар кетма-кет-лигини икки хил усулда ҳосил қилиш мумкин. Биринчи йўл - ҳар бир рейтинг учун алоҳида ўзгарувчи аниқлаш:  $Reyting_1, \dots, Reyting_N$ . Лекин, гуруҳдаги талабалар сони этарлича катта бўлганда, бу ўзгарув-чилар қатнашган программани тузиш катта қийинчиликларни юзага келтиради. Иккинчи йўл - берилганлар кетма-кетлигини ягона ном билан аниқлаб, унинг қийматларига мурожаатни, шу қийматларнинг кетма-кетликда жойлашган ўрнининг номери (индекси) орқали амалга оширишдир.

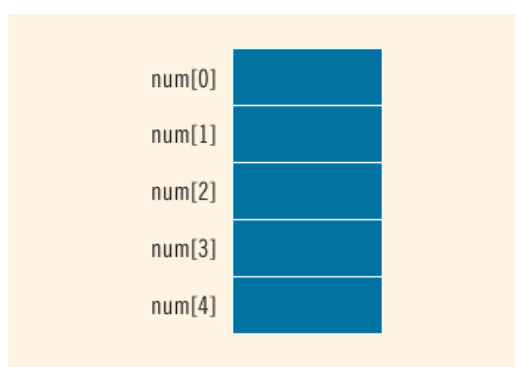
Рейтинглар кетма-кетлигини Рейтинг деб номлаб, ундаги қийматларига  $Reyting_1, \dots, Reyting_N$  кўринишида муружаат қилиш мумкин. Одатда берилганларнинг бундай кўринишига массивлар дейилади. Массивларни математикадаги сонлар векторига ўхшатиш мумкин, чунки вектор ҳам ўзининг индивидуал номига эга ва у фиксирланган миқдордаги бир турдаги қийматлардан - сонлардан иборатдир.

Демак, массив - бу фиксирланган миқдордаги айрим қийматлар-нинг (массив элементларининг) тартибланган мажмуасидир. Барча элементлар бир хил турда бўлиши керак ва бу тур элемент тури ёки массив учун таянч тур деб номланади. Юқоридаги келтирилган мисолда Рейтинг - ҳақиқий турдаги вектор деб номланади.

Программада ишлатиладиган ҳар бир конкрет массив ўзининг индивидуал номига эга бўлиши керак. Бу номни тўлиқ ўзгарувчи дейилади, чунки унинг қиймати массивнинг ўзи бўлади. Массивнинг ҳар бир элементи массив номи, ҳамда квадрат қавсга олинган ва элемент селектори деб номланувчи индексни кўрсатиш орқали ошкор равишда белгиланади.

### **Муружаат синтаксиси: `<massiv nomi >[<indeks>]`**

Бу кўринишга хусусий ўзгарувчи дейилади, чунки унинг қиймати массивнинг алоҳида элементи дир. Бизнинг мисолда Рейтинг массивининг алоҳида компоненталарига  $Reyting[1], \dots, Reyting[N]$  хусусий ўзгарув-чилар орқали муружаат қилиш мумкин. Бошқача бу ўзгарувчилар индексли ўзгарувчилар дейилади.

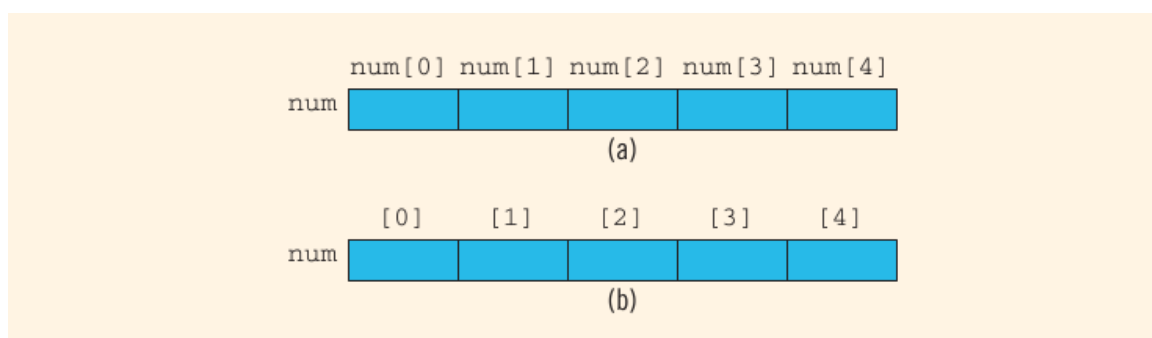


Массив индекси сифатида бутун сон қўлланилади. Умуман олганда индекс сифатида бутун сон қийматини қабул қиладиган ихтиёрий ифода ишлатилиши мумкин ва унинг қиймати массив элементи номерини аниқлайди. Ифода сифатида ўзгарувчи ҳам олиниши мумкинки, ўзгарувчининг қиймати

Ўзгариши билан муро-жаат қилинаётган массив элементини аниқловчи индекс ҳам ўзгаради. Шундай қилиб, программадаги битта индексли ўзгарувчи орқали массивнинг барча элементларини белгилаш (аниқлаш) мумкин бўлади. Масалан, `Reyting[I]` ўзгарувчиси орқали `i` ўзгарувчининг қийматига боғлиқ равишда Рейтинг массивининг ихтиёрий элементига мурожаат қилиш мавжуд.

## 11.2. Массив элементига мурожаат қилиш

Массивнинг элементларига мурожаат индекслари орқали бўлади. Индекс сифатида бутун турдаги ўзгарувчилардан фойдаланиш мумкин. Ҳақиқий турдаги (`float`, `double`) қийматлар тўплами чексиз бўлганлиги сабабли улар индекс сифатида ишлатилмайди.



C++ тилида индекс доимо 0 дан бошланади ва унинг энг катта қиймати массив эълонидаги узунликдан биттага кам бўлади.

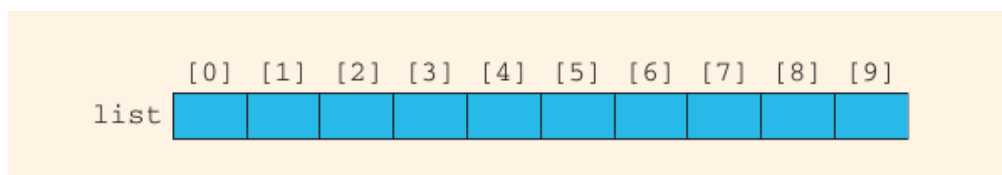
Массив эълони қуйидагича бўлади:

`<тур><ном> [<узунлик>] = {бошланғич қийматлар}.`

Бу ерда `<узунлик>` - ўзгармас ифода (константа).

Мисол: `int list[10];`

Бу ерда `list` номли массив элементлари 10 та бўлса, унинг элементлари `list[0]`, `list[1]`, `list[2]`, ... `list[9]` бўлади, бошқача айтганда массив элементлари 10 та:



Агар `list[5]=34;` бўлса

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
list						34				

34 қиймати массивнинг 5-элементига жойлаштирилади:

Бу ерда  $i$  бутун турдаги ўзгарувчи: `list[3]=63;`

Буни қуйидагича тушуниш мумкин  $i=3$ ; `list[i]=63;`

Агар  $i$  нинг қиймати 4 га тенг бўлсин, индекс ифада ҳам бўлиши мумкин:

`list[2*i-3]=58;`

бу ерда 58 қийматини массивнинг `list[5]` элементига юкланияпти чунки,  $2*i-3$  нинг қиймати 5 га тенг, операторда биринчи индекс ичидаги ифада бажарилади.

Қуйидаги мисолларни кўрайлик:

`list[3]=10;`

`list[6]=35;`

`list[5]= list[3]+list[6];`

Юқоридаги мисолда биринчи лист массивининг учинчи элементига 10 қийматини ўзлаштирияпти массивнинг олтинчи элементига 35 қийматини ўзлаштирияпти ва массивнинг учинчи ва олтинчи элементлари йиғиндиси массивнинг бешинчи элементига юкланияпти:

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
list				10		45	35			

Худди шунингдек массивни қуйидагича эълон қилиш мумкин:

`const int ARRAY_SIZE = 10;`

`int list[ARRAY_SIZE];`

бу ерда биринчи ўринда бутун турдаги ўзгармас эълон қилинияпти ва массив эълон қилиниб ўлчамлари ўрнатилияпти.

### 11.3. Кўп ўлчамли массивлар

C++ тилида массивлар элементининг турига чекловлар қўйилмайди, лекин бу турлар чекли ўлчамдаги объектларнинг тури бўлиши керак. Чунки компилятор массивнинг хотирадан қанча жой (байт) эгаллашини ҳисоблай олиши керак. Хусусан, массив компонентаси массив бўлиши мумкин («векторлар-вектори»), натижада матрица деб номланувчи икки ўлчамли массив ҳосил бўлади.

Агар матрицанинг элементи ҳам вектор бўлса, уч ўлчамли массивлар - куб ҳосил бўлади. Шу йўл билан эчилаётган масалага боғлиқ равишда ихтиёрий ўлчамдаги массивларни яратиш мумкин.

Икки ўлчамли массивнинг синтаксиси қуйидаги кўринишда бўлади:

```
<тур> <ном> [<узунлик >] [<узунлик>]
```

Масалан, 10x20 ўлчамли ҳақиқий сонлар массивининг эълони:

```
float a[10][20];
```

Эълон қилинган А матрицани кўриниши қуйидаги расмда келтирилган.

	[0]	[1]	[2]	[3]	[4]
[0]					
[1]					
[2]					
[3]					
[4]					
[5]					
[6]					
[7]					
[8]					
[9]					

$$\begin{array}{r}
 \mathbf{a}_0: (a_{00}, a_{02}, \dots \dots a_{018}, a_{019}), \\
 \mathbf{a}_1: (a_{10}, a_{11}, \dots \dots a_{118}, a_{119}), \\
 \dots \\
 \mathbf{i} \quad \mathbf{a}_i: (\dots, \dots, \dots \mathbf{a}_{ij} \dots \dots, \dots), \\
 \dots \\
 \mathbf{a}_9: (a_{90}, a_{91}, \dots \dots a_{918}, a_{919}).
 \end{array}$$

Энди адрес нуқтаи - назаридан кўп ўлчамли массив элемент-ларига мурожаат қилишни кўрайлик. Қуйидаги эълонлар берилган бўлсин:

```
int a[3][2];
float b[2][2][2];
```

Биринчи эълонда икки ўлчамли массив, яъни 2 сатр ва 3 устундан иборат матрица эълон қилинган, иккинчисидан уч ўлчамли - 3 та 2x2 матрицадан иборат бўлган массив эълон қилинган. Унинг элементларига мурожаат схемаси:



#### 11.4. Бир ўлчамли массивларни қайта ишлаш

Массивлар устида бажариладиган асосий амаллар берилганларни массив элементларига юклаш, массив элементлари устида амаллар бажариш ва массив элементларини чоп қилишдан иборат. Агар массив элементлари бутун сонлардан иборат бўлса унда массив элементларини йиғиндисини, ўрта арифметигинива бошқа амалларни бажариш мумкин. Бунда массивниг ҳар бир элементларига мурожаат қилишга тўғри келади, буни бошқариш осон. Мисол сифатида массив эълони қуйидагича бўлсин.

```
int list[100];
int i;
```



Қуйидаги такрорлаш оператори орқали массивнинг ҳар бир элементига муурожаат қилишимиз мумкин бўлади ва муурожаат массивнинг биринчи элементидан бошланади:

```
for (i = 0; i < 100; i++) . . .
```

Массив элементлари устида амаллар бажаришимиз учун берилганларни массивнинг ҳар бир элементларига ўқиб олишимиз керак бу син оператори орқали амалга оширилади. Мисол сифатида қуйидаги ифода массивнинг 100 та элементларини ўқиб олади:

```
for (i = 0; i < 100; i++) cin >> list[i];
```

Қуйида 9-3 мисолларда бир ўлчамли массив устида амаллар кўрсатилган, бизда массивнинг эълони қуйидагича бўлсин:

```
double sales[10];  
int index;  
double largestSale, sum, average;
```

Юқоридаги мисолда ҳақиқий турдаги 10 та элементдан ташкил топган салес массиви эълон қилинган. Бу массив устида қуйида амаллар бажарилган:

**a. Массивни инициализация қилиш:** Қуйида салес массивнинг ҳар бир элементига 0.0 қиймат билан инициализация қилинипти:

```
for(index=0; Index<10;index++) sales[index]=0.0;
```

**b. Массив элементларини ўқиб олиш:** Қуйидаги мисолда клавиатурадан киритилаётган берилганларни салес массивининг ҳар бир элементига ўқиб олинипти:

```
for (index = 0; index < 10; index++)  
    cin >> sales[index];
```

**c. Массив элементларини чоп қилиш:** Қуйидаги ифодада массивнинг ҳар бир элементи пробел белгиси ёрдамида ажратилиб чоп қилинипти:

```
for (index = 0; index < 10; index++)  
    cout << sales[index]<< " ";
```

**d. Массивнинг элементлари йиғиндисини ва массив элементларининг ўрта арифметигини топиш:** Бундай ҳолда бутун турдаги с ўзгарувчиси олиниб унга 0 бериб қўйилади ва такрорлаш оператори орқали

индекс ўзгарувчиси 0 дан n-1 гача ўзгаради. Массивнинг ҳар бир элементини с ўзгарувчисига йиғиб борамиз. Чиққан натижани 10 га бўлсак ўрта арифметиғи келиб чиқади;

```
int sum = 0;
for (index = 0; index < 10; index++)
    sum = sum + sales[index];
average = sum / 10;
```

**е. Массив элементларидан энг каттасини топишга доир:**Бу мисолда бутун турдаги maxIndex ўзгарувчисини олиб унга 0 бериб кўямиз ва такрорлаш операторини 1 дан бошлаймиз ва массивнинг нолинчи индексили элементи билан қолган ўриндаги элементлари билан солиштириб чиқамиз, агар ундан катта элемент чиқиб қолса maxIndex ўзгарувчисига катта элемент индексини оламиз ва текширишни давом эттирамиз. Текшириш натижасида бизга энг катта элемент индекси пайдо бўлади ва шу ўриндаги элементини чоп қиламиз;

```
maxIndex = 0;
for (index = 1; index < 10; index++)
    if (sales[maxIndex] < sales[index])
        maxIndex = index;
largestSale = sales[maxIndex];
```

Бу алгоритмни массивнинг қуйидаги қийматларида текшириб кўрайлик:

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
sales	12.50	8.35	19.60	25.00	14.00	39.43	35.90	98.23	66.65	35.64

Ҳар бир қадамда кўрамиз:

index	maxIndex	sales[maxIndex]	sales[index]	sales[maxIndex] < sales[index]
1	0	12.50	8.35	12.50 < 8.35 is false
2	0	12.50	19.60	12.50 < 19.60 is true; maxIndex = 2

3	2	19.60	25.00	19.60 < 25.00 is true; 1 maxIndex = 3
4	3	25.00	14.00	25.00 < 14.00 is false
5	3	25.00	39.43	25.00 < 39.43 is true; maxIndex = 5
6	5	39.43	35.90	39.43 < 35.90 is false
7	5	39.43	98.23	39.43 < 98.23 is true; maxIndex = 7
8	7	98.23	66.45	98.23 < 66.65 is false
9	7	98.23	35.64	98.23 < 35.64 is false-

for оператори бажарилгандан сўнг maxIndex=7 га тен бўлади ва бу массивнинг энг катта элементи жойлашган индекс ҳисобланади. Массивнинг бу индексидаги қиймати

```
largestSale = sales[maxIndex]= 98.23.
```

Биз массив эълон қилишни, массив элементларини ўқиб олишни массив элементларидан каттасини топишни ва элементлари йиғиндисини топиш алгоритмларини кўрдик. Энди C++ да дастурини ёзамиз:

```
#include <iostream>
using namespace std;
int main()
{
int item[5]; //Массив элементлари бешта бўлсин
int sum;
int counter;
cout << "Enter five numbers: ";
sum = 0;
for (counter = 0; counter < 5; counter++)
```

```

{
cin >> item[counter];
sum = sum + item[counter];}
cout << endl;
cout << "The sum of the numbers is: " << sum << endl;
cout << "The numbers in reverse order are: ";
//Қийматларни тескари тартибда чоп қилиш
for (counter = 4; counter >= 0; counter--)
cout << item[counter]<< " ";
cout << endl;
return 0;
}

```

Дастурни ишга туширамиз:

Бешта қийматни киритайлик: 12 76 34 52 89

Массив элементлари йиғиндиси: 263

Массив элементларини тескари тартибда чоп этиш: 89 52 34 76 12

### **11.5. Бир ўлчамли массивларни инициализациялаш**

Ихтиёрий массив эълонида унинг элементларига бошланғич қийматлар бериш мумкин. Бизга элементлари 5 та бўлган ҳақиқий турдаги сэлес массиви берилган бўлсин:

ўлчами кўрсатилган массив элементларини тўлиқ инициализациялаш:

```
double sales[5]= {12.25, 32.50, 16.90, 23, 45.68};
```

Бунда 5 та элементдан иборат бўлган сэлес номли ҳақиқий турдаги бир ўлчамли массив эълон қилинган ва унинг барча элементларига бошланғич қийматлар берилган.

Бунда массивнинг элементлари қуйидаги қийматларни ўзлаштиради:

```
sales[0] = 12.25, sales[1] = 32.50, sales[2] = 16.90,
sales[3] = 23.00, sales[4] = 45.68.
```

юқоридаги мисолда массивнинг инициализация қилишда массив ўлчами аниқланган, аммо массивнинг ўлчамини бермасдан ҳам уни инициализация қилиш мумкин, бунда массивнинг ўлчами берилган қийматлар сонидан аниқлаб олинади.

```
double sales[]= {12.25, 32.50, 16.90, 23, 45.68};
```

### **Массив элементларини тўлиқмас инициализация қилиш**

Массив элементларини эълон қилиш ва бир вақтда массивнинг барча элементларини инициализация қилиш мумкин, аммо массивни инициализация қилишда унинг қисман элементларини инициализация қилиш мумкин бунга массивни тўлиқмас инициализация қилиш дейилади.

```
Misol: int list[10]= {0};
```

10 та элементдан иборат бўлган массив эълон қилиняпти аммо массивнинг барча элементларига 0 қиймат инициализация қилиняпти;

```
int list[10]= { 8, 5, 12 };
```

10 та элементладан иборат бўлган массив эълон қилиняпти ва массивнинг қуйидаги элементлари list[0]=8, list[1]=5 va list[2]=12 қийматлар билан қолган элементлари 0 қиймат билан инициализация қилияпти. Агар массивнинг ўлчами кўрсатилмаган бўлса:

```
int list[ ]={5, 6, 3};
```

Юқоридаги мисолда массив элементлар сони келтирилмаган бунда ҳолда массив элементлари сони инициализация қилинаётган қийматлар сони билан аниқлаб олинади, элементлар сони учта ва массивнинг list[0] элемент 5 ни, list[1] элемент 6 ни va list[2] элемент 3 ни олади.

```
int list[25]={4 , 7};
```

Бундай ҳолда массивнинг биринчи иккита элементлари 4 ва 7 қийматларни қолган элементлар 0 ни инициализация қилади.

```
int t[5]={-10,5,15};
```

Бу ерда фақат массив бошидаги учта элементга бошланғич қийматлар берилган. Шуни айтиб ўтиш керакки, массивнинг бошидаги ёки ўртасидаги элементларига қийматлар бермасдан, унинг охиридаги элементларга бошланғич қиймат бериш мумкин эмас. Агарда массив элементларига

бошланғич қиймат берилмаса, унда келишув бўйича статис ва эхтерн модификатори билан эълон қилинган массив учун элементларининг қиймати 0 сонига тенг деб, аутоматис массивлар элементларининг бошланғич қийматлари номаълум ҳисобланади.

```
int t[]={-10,5,15,4,3};
```

Бу мисолда массивни барча элементларига қийматлар берилган ҳисобланади, массив узунлиги компилятор томонидан бошланғич қийматлар сонига қараб аниқланади. Агарда массив узунлиги берилмаса, бошланғич қиймати берилиши шарт.

Массивда мусбат элеменлар сони ва суммасини ҳисоблаш.

```
#include <iostream>
#include <conio.h>
using namespace std;
void main()
{
int x[]={-1;2;5;-4;8;9};
clrscr();
for (int s=0,int k=0, int I=0; I<6; I++)
{
if (x[I]<=0)
continue;
k++;
s++;
};
cout << ("%d",k);
cout << ("%d",k);
getch();
}
```

Массивнинг энг катта, энг кичик элементи ва ўрта қийматини аниқлаш:

```
#include <iostream>
using namespace std;
void main()
{
int I,j,n;
float a,b,d,x[100];
while(1)
{
cout << (“\n n=”);
cin >> (“%i”,&n);
if ( n > 0 && n < = 100 ) break;
cout << (“\n Xato 0<n<101 bulishi kerak”);
}
cout << (“\n elementlar kiymatlarini kiriting:\n”);
For (i=0; i<n; i++)
{
cout << (“x[%i]=”,i);
cin >> (“%f”,&x[i]);}
max=x[0];
min=x[0];
for (s=0,i=0;i<n;i++)
{
s++;
if (max<x[i]) max=x[i];
if (min>x[i]) min=x[i];
};
```

```

s /= n;
cout << (“\n max=%f”,max);
cout << (“\n min=%f”,min);
cout << (“\n urta kiyamat=%f”,s);
}

```

### 11.6. Кўп ўлчамли массивларни инициализациялаш

Массивларни инициализациялаш қуйидаги мисолларда кўрсатилган:

```

int a[2][3]={0,1,2,10,11,12};
int b[3][3]={{0,1,2},{10,11,12},{20,21,22}};
int c[3][3][3]={{0},{100,101},{110}},
{{200,201,202},{210,211,212},{220,221,222}};

```

Биринчи операторда бошланғич қийматлар кетма-кет ёзилган, иккинчи операторда қийматлар гуруҳлашган, учинчи операторда ҳам гуруҳлашган, лекин баъзи гуруҳларда охириги қийматлар берилмаган.

Икки улчовли массивлар математикада матрица ёки жадвал тушунчасига москелади. Жадвалларни инициализация қилиш қоидаси, икки ўлчовли массивнинг элементлари массивлардан иборат бўлган бир ўлчовли массив таърифига асослангандир. Мисол учун икки қатор ва уч устундан иборат бўлган ҳақиқий типга тегишли д массив бошланғич қийматлари қуйидагича кўрсатилиши мумкин:

```
float d[2][3]={{(1, -2.5, 10), (-5.3, 2, 14)}};
```

Бу ёзув қуйидаги қиймат бериш операторларига мосдир:

```

d[0][0]=1;
d[0][1]=-2.5;
d[0][2]=10;
d[1][0]=-5.3;
d[1][1]=2;
d[1][2]=14;

```

Бу қийматларни битта рўйхат билан ҳосил қилиш мумкин:

```
float d[2][3]={1, -2.5, 10, -5.3, 2, 14};
```



Инициализация ёрдамида бошланғич қийматлар аниқланганда массивнинг ҳамма элементларига қиймат бериш шарт эмас.

Мисол учун: `int x[3][3]={(1, -2, 3), (1, 2), (-4)}`.

Бу ёзув қўйидаги қиймат бериш операторларига мосдир:

```
x[0][0]=1;
```

```
x[0][1]=-2;
```

```
x[0][2]=3;
```

```
x[1][0]=-1;
```

```
x[1][1]=2;
```

```
x[2][0]=-4;
```

Инициализация ёрдамида бошланғич қийматлар аниқланганда массивнинг биринчи индекси чегараси кўрсатилиши шарт эмас, лекин қолган индекслар чегаралари кўрсатилиши шарт.

Мисол учун:

```
double x[][2]={(1.1, 1.5), (-1.6, 2.5), (3, -4)}
```

Бу мисолда автоматик равишда қаторлар сони учга тенг деб олинади.

Қўйидаги кўрадиган мисолимизда жадвал киритилиб ҳар бир қаторнинг максимал элементи аниқланади ва бу элементлар орасида энг кичиги аниқланади:

```
#include <iostream>
using namespace std;
void main()
{
double a[4,3]; double s,max=0.0,min=0.0;
int i,j;
for (i=0;i<4;i++) {
for (j=0;j<3;j++)
{
```

```

cout << (" a[%d][%d]=",i,j);
cin >> ("%f",s);a[i,j]=s;
if (max<s) max=s;
};
cout << ("\n");
if (max<min) min=max;
}
cout << ("\n min=%f", min);
}

```

Мисол учун, матрицалар ва вектор кўпайтмасини –  $C=A*b$  ни ҳисоблаш масаласини кўрайлик. Бу ерда  $A=\{a_{ij}\}$ ,  $b=\{b_j\}$ ,  $C=\{c_{ij}\}$ ,  $0 \leq i < m$ ,  $0 \leq j < n$ .

Ҳисоблаш формуласи:

$$c_i = \sum_{j=0}^{n-1} a_{ij} b_j$$

Мос програма матни:

```

void main()
{
    const int n=4,m=5;
    float a[m][n],b[n],c[m];
    int i,j; float s;
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)cin>>a[i][j];
    for(i=0;i<m;i++)cin>>b[i];
    for(i=0;i<m;i++)
    {
        for (j=0,s=0;j<n;j++)s+=a[i,j]*b[j];
        c[i]=s;
    }
}

```

```

}
for (i=0;i<m;i++)cout<<"\t c["<<i<<"]="<<c[i];
return;
}

```

Символли массивлар

С ++ тилида сатрлар символли массивлар сифатида таърифланади. Символли массивлар қўйидагича тасвирланиши мумкин:

```
char pas[10];
```

Символли массивлар қўйидагича инициализация қилинади:

```
char capital[]="TASHKENT";
```

Бу ҳолда автоматик равишда массив элементлари сони аниқланади ва массив охирига сатр кўчириш ъ\нъ символи кўшилади.

Юқоридаги инициализацияни қўйидагича амалга ошириш мумкин:

```
char capital[]={ 'T', 'A', 'S', 'H', 'K', 'E', 'N', 'T', '\n' };;
```

Бу ҳолда сўз охирида ‘\’ символи аниқ кўрсатилиши шарт.

Мисол учун палиндром масаласини кўриб чиқамиз. Палиндром деб олдидан ҳам охиридан ҳам бир хил ўқиладиган сўзларга айтилади. Мисол учун нон. Дастурда киритилган сўз палиндром эканлиги аниқланади:

```

#include <iostream>
using namespace std;
void main()
{
gets(a);
for ( int j=0, a[j]!='\0';j++);
I=0;
while (I<j) if (a[I++]!=a[j--]) break;
if ((j-I)>1)
cout << ("Palindrom emas");
}

```

```
else cout<<("Palindrom");
```

Кейинги мисолимизда киритилган сўздан берилган ҳарф олиб ташлаш дастури берилган:

```
#include <iostream>
using namespace std;
void main()
{
char s[];
int c;
gets(a);
int i, j;
for ( i = j = 0; s[i] != '\0'; i++)
if ( s[i] != c )
s[j++] = s[i];
s[j] = '\0';
puts(s);
}
```

Ҳар гал 's' дан фарқли символ учраганда , у j позитсияга ёзилади ва фақат шундан сўнг j қиймати 1 га ошади. Бу қўйидаги ёзувга эквивалент:

```
if ( s[i] != c )
s[j] = s[i];
j++;
```

C ++ тилида сўзлар массивлари икки ўлчовли символли массивлар сифатида таърифланади. Мисол учун:char Name[4][5].

Бу таъриф ёрдамида ҳар бири 5 та ҳарфдан иборат бўлган 4 та сўзли массив киритилади. Сўзлар массивлари қўйидагича инициализация қилиниши мумкин:

```
char Name[3][8]={“Anvar”,”Mirkomil”,”Yusuf”}.
```

Бу таърифда ҳар бир сўз учун хотирадан 8 байт жой ажратилади ва ҳар бир сўз охирига '\0' белгиси қўйилади.

Сўзлар массивлари инициализация қилинганда сўзлар сони кўрсатилмаслиги мумкин. Бу ҳолда сўзлар сони автоматик аниқланади:

```
char comp[][9]={"komp'yuter","printer","kartridj"}.
```

Қуйидаги дастурда берилган ҳарф билан бошланувчи сўзлар руйҳати босиб чиқарилади:

```
#include <iostream>
using namespace std;
void main()
{
char a[10][10];
char c;
for (int i=0;i<10;i++)
    gets(a[i]);
c=getchar();
for (i=0;i<10;i++)
    if (a[i][0]==c)
        puts(a[i]);
}
```

Қуйидаги дастурда фан номи, талабалар руйҳати ва уларнинг баҳолари киритилади. Дастур бажарилганда икки олган талабалар руйҳати босиб чиқарилади:

```
#include <iostream>
using namespace std;
void main()
{
char a[10][10];
```

```

char s[10];
int k[10];
gets(s);
for (int i=0;i<10;i++)
    gets(a[i]);
for (i=0;i<10;i++)
    cin >> ("%d",k[i]);
for (int i=0;i<10;i++)
    if (k[i]==2)
        puts(a[i]);
}

```

#### **Адабиётлар рўйхати:**

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

#### **Мустақил таълим мавзулари:**

1. Векторлар устида амаллар бажариш.
2. Массивларни тасодифий сонлар билан тўлдириш усуллари.
3. Матрицалар устида амаллар бажариш.
4. Массивларни тасодифий сонлар билан тўлдириш.

## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур массивдаги иккиннинг даражаси бўладиган элементларни индексларини чоп этади. Лекин дастур ўз ишини тўхтатмасдан давом этмоқда. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур коддини тақдим этинг (индивидуал ҳолда).

### Назорот саволлари:

1. Массив деб нимага айтилади?
2. Массив индекси сифатида қандай сон ишлатилади?
3. Програмада ишлатилган ҳар бир конкрет массив қандай номга ега?
4. Массив элементига мурожаат қилиш қандай амалга оширилади?
5. C++ тилида массивлар элементининг турига чекловлар қуйиладими?
6. Икки ўлчамли массивнинг синтаксиси қандай қуринишда бўлади?
7. Сўзлар массивлари инициализация қилинганда сўзлар сони кўрсатилмаслиги мумкин. Бу ҳолда сўзлар сони қандай аниқланади?
8. Include оператори қандай вазифани бажаради?
9. Cout оператори қандай вазифани бажаради?
10. Cin оператори қандай вазифани бажаради?
11. For оператори қандай вазифани бажаради?
12. While оператори қандай вазифани бажаради?
13. do-while оператори қандай вазифани бажаради?
14. Char оператори қандай вазифани бажаради?
15. Iostream оператори қандай вазифани бажаради?
16. Массив – бу?
17. Палиндром деб нимага айтилади?

18. мисолда массив элементлар сони келтирилмаган булса массив элементлар сони кандай аникланади?

19. Int оператори кандай вазифани бажаради?

20. Gets оператори кандай вазифани бажаради?

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шархи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади.
cout		экранга чиқариш оқими.
cin		қийматларни ўқиб олиш
for		такрорлаш оператори
while		шарти олдин келадиган такрорлаш оператори
do-while		шарти кейин келадиган такрорлаш оператори
char		ўзгарувчиларнинг белги тури
iostream		киритиш чиқариш оқимлари билан ишлайдиган кутубхона

### Тест саволлари:

1. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int a[]={10,11,12,13,14};  
int s=0;  
  
for (int i=0;i<4;i++)  
s=s+a[i];  
cout << s;
```



- a) 46
- b) 55
- c) 21
- d) 60

2. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int a[]={10,11,12,13,14};
int s=0;
for (int i=0;i<4;i++)
if (a[i]%2) s=s+a[i];
cout << s;
```

- a) 22
- b) 46
- c) 60
- d) 24

3. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int a[]={-20,30,-40,43,64};
int s=6;
for (int i=0;i<5;i++)
if (a[i]<0) s=s+a[i];
```

- a) -54
- b) 60
- c) -60
- d) -24

4. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int a[]={-20,30,-40,43,64};
int s=0;
for (int i=0;i<5;i++)
if (a[i]<0) s=s+a[0];
cout << s;
```

- a) -20
- b) -40
- c) -60
- d) 40

5. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int a[]={-20,30,-40,43,64};  
int s=0;  
for (int i=0;i<3;i++)  
    if (a[i]>0) s=s+a[i];  
s=s+a[2];  
cout << s;
```

- a) -10
- b) 30
- c) -70
- d) 40

6. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
double a[]={2.5, 3.9, 4.8, 6.2};  
int s=0;  
for (int i=0;i<4;i++)  
    s=s+a[i];  
cout << s;
```

- a) 15
- b) 17.4
- c) 16.4
- d) 15.4

7. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
double a[]={2.5, 3.9, 4.8, 6.2};  
int m = 0;  
for (int i = 1; i < 4; i++)  
if (a [m] < a [i])  
m = i;  
cout << a [m];
```

- a) 6.2
- b) 4.8
- c) 3.9
- d) 2.5

8. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
double a[]={2.5, 3.9, 4.8, 6.2};  
double s=0;  
for ( int i=0;i<4;i++)  
s=s+a[i];  
cout << s;
```

- a) 17.4
- b) 15
- c) 16.4
- d) 15.4

9. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
double a[]={2.5, 3.9, 4.8, 6.2};  
int m = 0;  
for ( int i = 1; i < 4; i++)  
if (a [m] > a [i])  
m = i;  
cout << a [m];
```

- a) 2.5
- b) 4.8
- c) 3.8
- d) 6.2

10. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int list[] = {6, 8, 2, 14, 13};  
for (int i = 0; i < 4; i++)  
list[i] = list[i] - list[i + 1];  
for (int i = 0; i < 5; i++)  
cout << list[i];
```

- a) -26-12113
- b) 6821413
- c) 2612311
- d) 3141286

11. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int list[] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
int s=0, j=0;  
for (int i = 0; i < 3; i++)  
s=s+a[i][j];  
cout << s;
```

- a) 12
- b) 15
- c) 45
- d) 6

12. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
double a[]={12.53, 23.49, 4.8, 6.29};  
int m = 0;  
for ( int i = 1; i < 4; i++)  
if (a [m] > a [i])  
m = i;  
cout << m;
```

- a) 2
- b) 1
- c) 3
- d) 0

13. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int list[] ={{1,2,3},{4,5,6},{7,8,9}};  
int s=0, j=1;  
for (int i = 0; i < 3; i++)  
s=s+a[j][i];  
cout << s;
```

- a) 15
- b) 6
- c) 12
- d) 45

14. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int list[] ={{1,2,3},{4,5,6},{7,8,9}};  
int s=0;  
for (int i = 0, j=0; i < 3; i++, j++)  
s=s+a[i][j];  
cout << s;
```

- a) 15
- b) 6
- c) 45
- d) 12

15. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
double a[]={12.53, 23.49, 4.8, 6.29};  
int m = 0;  
for ( int i = 1; i < 4; i++)  
if (a [m] < a [i])  
m = i;  
cout << m;
```

- a) 1
- b) 2
- c) 3
- d) 0

16. Қайси жавобда массив эълони тўғри берилган?

- a) double d[]={2,4,6,4}
- b) double d={2,4,6,4}
- c) double d[2]={2,4,6,4}
- d) double d[]={2, ,6,4}

17. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int list[] ={{1,2,3},{4,5,6},{7,8,9}};  
int s=0;  
for (int i = 0; i < 3; i++)  
for (int j = 0; j < 3; j++)  
s=s+a[i][j];  
cout << s;
```

a) 45

b) 6

c) 15

d) 12

18. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int list[] ={{21,2,43}, {31,5,26}, {7,18,22}};  
int s=0, min=0;  
for (int i = 0; i < 3; i++)  
for (int j = 0; j < 3; j++)  
if (min>a[i][j]) min=a[i][j]  
cout << min;
```

a) 0

b) 2

c) min

d) 43

19. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int list[] ={{21,2,43}, {31,5,26}, {7,18,22}};  
int s=0, min=a[0][0];  
for (int i = 0; i < 3; i++)
```

```
for (int j = 0; j < 3; j++)  
if (min<a[i][j]) min=a[i][j]  
cout << min;
```

- a) 43
- b) 2
- c) min
- d) 0

20. Қуйидаги дастур қисми ишлаши натижасида экранга нима чиқади?

```
int list[] ={{21,2,43}, {31,5,26}, {7,18,22}};  
int s=0, min=j=2;  
for (int i = 0; i < 3; i++)  
if (min>a[i][j]) min=a[i][j]  
cout << min;
```

- a) 2
- b) 22
- c) min
- d) 7



## Маъруза 13-14. Функциялар эълон қилиш ва аниқлаш. main() функцияси. Локал ва глобал ўзгарувчилар

### Режа:

1. Функциялардан фойдаланиш
2. Олдиндан аниқланган функциялар
3. Фойдаланувчи томонидан аниқланадиган функциялар
4. Келишув бўйича аргументлар.
5. Кўриниш соҳаси. Локал ва глобал ўзгарувчилар. :: амали
6. Жойлаштириладиган (inline) функциялар
7. Қайта юкланувчи функциялар

**Таянч атамалар:** *include, Функция, Параметр, аргумент, сарлавҳа файли, cout, cin, if, Амал қилиш соҳаси, Ўзгарувчи, Идентификатор, else.*

### 13.1. Функциялардан фойдаланиш

Дастур таъминотини яратиш амалда мураккаб жараён ҳисобланади. Дастур тузувчи дастур комплексини бир бутун-ликдаги ва унинг ҳар бир бўлагининг ички мазмунини ва уларнинг сезилмас фарқларини ҳисобга олиши керак бўлади.

Дастурлашга тизимли ёндошув шундан иборатки, дастур тузувчи олдида кўйилган масала олдиндан иккита, учта ва ундан ортиқ нисбатан кичик масала остиларга бўлинади. Ўз навбатида бу масала остилари ҳам яна кичик масала остиларига бўлиниши мумкин. Бу жараён токи майда масалаларни оддий стандарт амаллар ёрдамида ечиш мумкин бўлгунча давом этади. Шу йўл билан масалани декомпозициялаш амалга оширилади.

Иккинчи томондан, дастурлашда шундай ҳолатлар кузатиладики, унда дастурнинг турли жойларида мазмунан бир хил алгоритмларни бажаришга тўғри келади. Алгоритмнинг бу бўлаклари асосий эчилаётган масаладан ажратиб олинган қандайдир масала остини ечишга мўлжалланган бўлиб, этарлича мустақил қийматга (натижага) эгадир. Мисол учун қуйидаги масалани кўрайлик:

Берилган  $a_0, a_1, \dots, a_{30}, b_0, b_1, \dots, b_{30}, c_0, c_1, \dots, c_{30}$  ва  $x, y, z$  ҳақиқий сонлар учун

$$\frac{(a_0x^{30} + a_1x^{29} + \dots + a_{30})^2 - (b_0y^{30} + b_1y^{29} + \dots + b_{30})}{c_0(x+z)^{30} + c_1(x+z)^{29} + \dots + c_{30}}$$

ифоданинг қиймати ҳисоблансин.

Бу мисолни ечишда касрнинг сурат ва махражидаги ифодалар бир хил алгоритм билан ҳисобланади ва дастурда ҳар бир ифодани (масала ости) ҳисоблаш учун бу алгоритмни 3 марта ёзишга тўғри келади. Масаладаги 30-даражали кўпхадни ҳисоблаш алгоритмини, масалан, Горнер алгоритмини алоҳида, битта нусхада ёзиб, унга турли параметрлар - бир сафар  $a$  вектор ва  $x$  қийматини, иккинчи сафар  $b$  вектор ва  $y$  қийматини, ҳамда  $c$  вектор ва  $(x+z)$  қийматлари билан мурожаат қилиш орқали асосий масалани ечиш мумкин бўлади. Функциялар қўлланишининг яна бир сабабини қуйидаги масалада кўришимиз мумкин - берилган чизиқли тенгламалар системасини Гаусс, Крамер, Зейдел усулларининг бирортаси билан ечиш талаб қилинсин. У ҳолда асосий дастурни қуйидаги бўлақларга бўлиш мақсадга мувофиқ бўлар эди: тенглама коэффитцентларини киритиш бўлаги, ечиш усулини танлаш бўлаги, Гаусс, Крамер, Зейдел усулларини амалга ошириш учун алоҳида бўлақлар, натижани чоп қилиш бўлаги. Ҳар бир бўлақ учун ўз функциялар мажмуаси яратиб, зарур бўлганда уларга бош функция танасидан мурожаатни амалга ошириш орқали бош масала ечиш самарали ҳисобланади.

Бундай ҳолларда дастурни ихчам ва самарали қилиш учун C++ тилида дастур бўлагини алоҳида ажратиб олиб, уни функция кўринишида аниқлаш имкони мавжуд.

Функция бу – C++ тилида масала ечишдаги калит элементларидан биридир. Функциялар модуллар деб ҳам аталади. Функциялар олдиндан аниқланган ва фойдаланувчи томонидан аниқланадиган функцияларга бўлинади.

### 13.2. Олдиндан аниқланган функциялар

Олдиндан аниқланган функциялар асосан тилнинг турли кутубхона файллари орқали аниқланган. Уларга математик функциялар, турларни

текшириш функциялари, белги ва сатрлар билан ишлаш функциялари мисол бўлади. Масалан:

Функция ишлатилиши	Кутубхона файли	Бажарадиган амали
abs(x)	<cmath>	x бутун сонининг абсолют қийматини қайтаради
fabs(x)	<cmath>	x ҳақиқий сонининг абсолют қийматини қайтаради
log(x)	<cmath>	x сонининг натурал логарифмини қайтаради
pow(x, y)	<cmath>	$x^y$ ҳисоблайди
sqrt(x)	<cmath>	x сонининг квадрат илдизини қайтаради
islower(x)	<cctype>	x қийматини кичик ҳарфлигини текширади
isupper(x)	<cctype>	x қийматини катта ҳарфлигини текширади
tolower(x)	<cctype>	x қийматини кичик ҳарф кўринишига айлантиради
toupper(x)	<cctype>	x қийматини катта ҳарф кўринишига айлантиради

### 13.3. Фойдаланувчи томонидан аниқланадиган функциялар

Дастурда ишлатиладиган ҳар қандай фойдаланувчи томонидан аниқланадиган функциялар эълон қилиниши керак. Функциялар қиймат қайтарувчи ва қиймат қайтармайдиган функциялар кўринишида бўлади.

Одатда функциялар эълони сарлавҳа файлларда эълон қилинади ва `#include` директиваси ёрдамида дастур матнига қўшилади.

Функция эълонини *функция прототипи* тавсифлайди (айрим ҳолларда *сигнатура* дейилади). Функция прототипи куйидаги кўринишда бўлади:

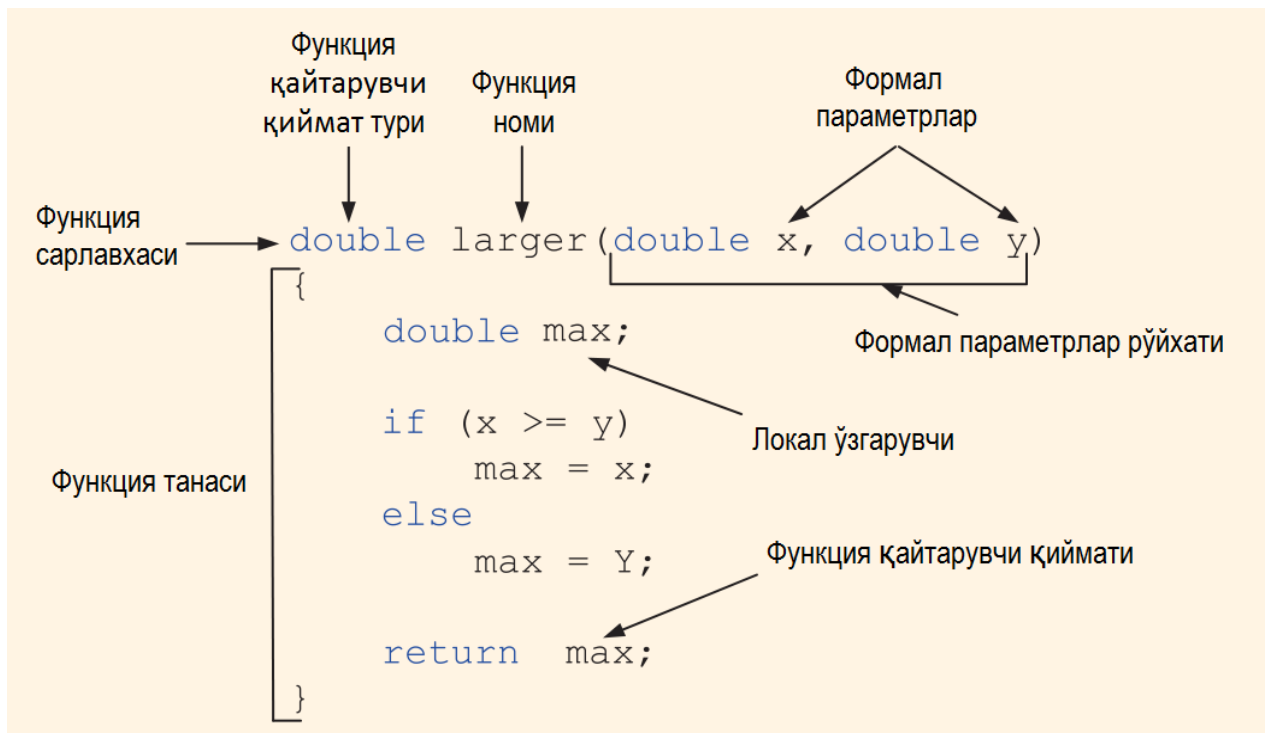
<қайтарувчи қиймат тури> <функция номи>(<параметрлар рўйхати >);

Бу ерда <қайтарувчи қиймат тури> - функция ишлаши натижасида у томонидан қайтарадиган қийматнинг тури. Агар қайтариладиган қиймат тури кўрсатилмаган бўлса, келишув бўйича функция қайтарадиган қиймат тури инт деб ҳисобланади, <параметрлар рўйхати>- вергул билан ажратилган функция параметрларининг тури ва номлари рўйхати. Параметр номини ёзмаса ҳам бўлади. Рўйхат бўш бўлиши ҳам мумкин. Функция прототипларига мисоллар:

```
int almashsin(int, int);  
double max(double x, double y);  
void func();  
void chop_etish(void);
```

Функция прототипи тушириб қолдирилиши мумкин, агар дастур матнида функция аниқланиши уни чақирадиган функциялар матнидан олдин ёзилган бўлса. Лекин бу ҳолат яхши услуб ҳисобланмайди, айниқса ўзаро бири-бирига мурожаат қилувчи функцияларни эълон қилишда муаммолар юзага келиши мумкин.

*Функция аниқланиши* - функция сарлавҳаси ва фигурали қавсга ('{, }') олинган қандайдир амалий мазмунга эга танадан иборат бўлади. Агар функция қайтарувчи тури воид туридан фарқли бўлса, унинг танасида албатта мос турдаги параметрга эга ретурн оператори бўлиши шарт. Функция танасида биттадан ортиқ ретурн оператори бўлиши мумкин. Уларнинг ихтиёрий бирортасини бажариш орқали функциядан чиқиб кетилади. Агар функциянинг қиймати дастурда ишлатилмайдиган бўлса, функциядан чиқиш учун параметрсиз ретурн оператори ишлатилиши мумкин ёки умуман ретурн ишлатилмайди. Охирги ҳолда функциядан чиқиш - охирги ёпилувчи қавсга этиб келганда рўй беради.



Функция дастурнинг бирорта модулида ягона равишда аниқланиши керак, унинг эълони эса функцияни ишлатадиган модулларда бир неча марта ёзилиши мумкин. Функция аниқланишида сарлавҳадаги барча параметрлар номлари ёзилиши шарт.

Одатда дастурда функция маълум бир ишни амалга ошириш учун чақирилади. Функцияга мурожаат қилганда, у қўйилган масалани эчади ва ўз ишини тугатишида қандайдир қийматни натижа сифатида қайтаради.

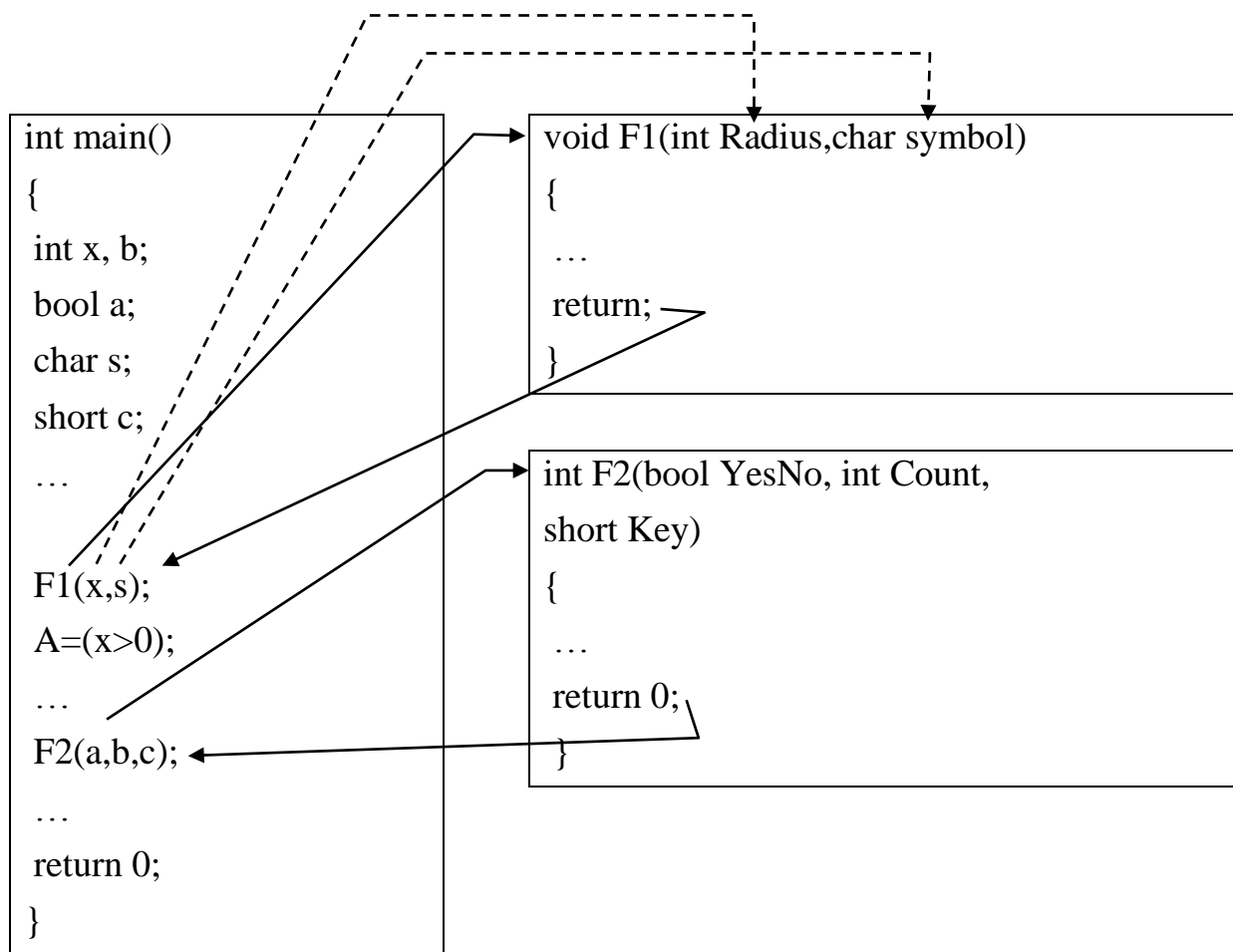
*Функцияни чақириши* учун унинг номи ва ундан кейин кавс ичида аргументлар рўйхати берилади:

<функция номи>( <аргумент1>, <аргумент2>, ..., <аргументн > );

Бу ерда ҳар бир <аргумент> - функция танасига узатиладиган ва кейинчалик ҳисоблаш жараёнида ишлатиладиган ўзгарувчи, ифода ёки ўзгармасдир. Аргументлар рўйхати бўш бўлиши мумкин.

Олдинги бобларда таъкидлаб ўтилганидек, C++ тилидаги ҳар қандай дастурда албатта main() бош функцияси бўлиши керак. Айти шу функцияни юклагич томонидан чақирилиши билан дастур бажарилиши бошланади.

Қуйидаги расмда бош функциядан бошқа функцияларни чақириш ва улардан қайтиш схемаси кўрсатилган.



Дастур `main()` функциясини бажаришдан бошланади ва «**F1(x,s);**» - функция чақиришгача давом этади ва кейинчалик бошқарув **F1()** функция танасидаги амалларни бажаришга ўтади. Бунда **Radius** параметрининг қиймати сифатида функция **x** ўзгарувчи қийматини, **symbol** параметри сифатида **s** ўзгарувчисининг қиймати ишлатилади. Функция танаси **return** операторигача бажарилади. **return** оператори бошқарувни `main()` функцияси танасидаги **F1()** функцияси чақирилган оператордан кейинги операторга ўтишни таъминлайди, яъни функциядан қайтиш рўй беради. Шундан кейин `main()` функцияси операторлари бажарилишда давом этади ва «**F2(a,b,c)**» - функция чақириши орқали бошқарув **F2()** функция танасига ўтади ва ҳисоблаш жараёнида мос равишда **YesNo** сифатида **a** ўзгарувчисининг, **Count** сифатида **b** ўзгарувчисининг ва **Key** сифатида **c** ўзгарувчисининг қийматлари ишлатилади. Функция танасидаги **return** оператори ёки охириги оператор бажаргандан кейин автоматик равишда бош функцияга қайтиш амалга оширилади.

Аксарият ҳолларда main() функциясининг параметрлар рўйхати бўш бўлади. Агар юкланувчи дастурни ишга туширишда, буйруқ сатри орқали юкланувчи дастур ишга туширилганда, унга параметрларни узатиш (бериш) зарур бўлса, main() дастури функциясининг синтаксиси ўзгаради:

```
int main(int argc, char* argv[]);
```

Бу ерда argc - узатиладиган параметрлар сони, argv[] – бир-биридан пунктуация белгилари (ва пробел) билан ажратилган параметрлар рўйхатини ўз ичига олган массивга кўрсаткич.

Қуйида функцияларни эълон қилиш, чақириш ва аниқлашга мисоллар келтирилган:

```
// funksiyalar e'loni
int Mening_funksiyam(int Number, float Point);
char Belgini_uqish();
void bitni_urnatish(short Num);
void Amal_yoq(int, char);
// funksiyalarni chaqirish
result = Mening_funksiyam(Varb1, 3.14);
symb = Belgini_uqish();
bitni_urnatish(3);
Amal_yoq(2, Smb1);
// funksiyalarni aniqlash
int Mening_funksiyam(int Number, float Point);
{
int x;
...
return x;
}
char Belgini_uqish()
```

```

{
char Symbol;
cin >> Symbol;
return Symbol;
};
void bitni_urnatish(short number)
{
global_bit = global_bit | number;
};
void Amal_yoq(int x, char ch){};

```

Функциянинг дастурдаги ўрнини янада тушунарли бўлиши учун сон квадратини ҳисоблаш масаласида функциядан фойдаланишни кўрайлик.

Функция прототипини sarlavha.h сарлавҳа файлида жойлаштирамиз:

```
long Son_Kvadrati(int);
```

Асосий дастурга ушбу сарлавҳа файлини қўшиш орқали Son\_Kvadrati() функция эълони дастур матнига киритилади:

```

#include <iostream>
using namespace std;
#include "sarlavha.h"
int main()
{
int Uzgaruvchi=5;
cout << Son_Kvadrati(Uzgaruvchi);
return 0;
}
long Son_Kvadrati(int x) {return x*x;}

```



Худди шу масалани сарлавҳа файлидан фойдаланмаган ҳолда, функция эълонини дастур матнига ёзиш орқали ҳам ҳал қилиш мумкин:

```
#include <iostream>
using namespace std;
long Son_Kvadrati(int);
int main()
{
int Uzgaruvchi = 5;
cout << Son_Kvadrati(Uzgaruvchi);
return 0;
}
long Son_Kvadrati(int x){ return x*x;}
```

Дастур ишлашида ўзгариш бўлмайти ва натижа сифатида экранга 25 сонини чоп этади.

**Масала.** Иккита туб сон «егизак» дейилади, агар улар бир-биридан 2 фарқ қилса (масалан, 41 ва 43 сонлари). Берилган натурал  $n$  учун  $[n..2n]$  кесмадаги барча «егизак» сонлар жуфтликлари чоп этилсин. Масалани ечиш учун берилган  $k$  сонини туб сон ёки йўқлиги аниқловчи мантиқий функцияни тузиш зарур бўлади. Функцияда  $k$  сони  $2..k/2$  гача сонларга бўлинади, агар  $k$  бу сонларнинг бирортасига ҳам бўлинмаса, у туб сон ҳисобланади ва функция true қийматини қайтаради. Бош функцияда, берилган  $n$  учун  $[n..2n]$  ораликдаги  $(n, n+2)$ ,  $(n+1, n+3)$ , ...,  $(2n-2, 2n)$  сон жуфтликларини туб сонлар эканлиги текширилади ва шартни қаноатлантирган жуфтликлар чоп этилади.

Дастур матни:

```
bool TubSon(unsigned long k);
int main()
{
unsigned long n,i;
unsigned char egizak=0;
```

```

cout<<"n -> ";
cin>>n;
cout<< '['<<n<<".."<<2*n<<']';
for(i=n; i<=2*n-2; i++)
    if(TubSon(i) && TubSon(i+2))
    {
        if (!egizak)
            cout<<" oralig'idagi egizak tub sonlar:\n";
        else cout<<" ";
        egizak=1;
        cout<< '{'<<i<<','<<i+2<<'}';
    };
if(!egizak)
    cout<<" oralig'ida egizak tub sonlar mavjud emas.";
else cout<<'.';
return 0;
}
bool TubSon(unsigned long k)
{
    unsigned long m;
    for (m=2; m<=k/2; m++)
        if (k%m==0) return false;
    return true;
}

```

Натурал  $n$  сони учун 100 киритилса, дастур қуйидаги сонлар жуфтликларини чоп қилади:

[100..200] oralig'idagi egizak tub sonlar:

{101,103}; {107,109}; {137,139}; {149,151};  
{179,181}; {191,193}; {197,199}.

### 13.4. Келишув бўйича аргументлар.

C++ тилида функция чакирилганда айрим аргументларни тушириб қолдириш мумкин. Бунга функция прототипода ушбу параметрларни келишув бўйича қийматини кўрсатиш орқали эришиш мумкин. Масалан, қуйида прототипи келтирилган функция турли чакиришга эга бўлиши мумкин:

```
//funksiya prototipi  
void Butun_Son(int I, bool Bayroq=true, char Blg='\n');  
  
//funksiyani chaqirish variantlari  
Butun_Son(1, false, 'a');  
Butun_Son(2, false);  
Butun_Son(3);
```

Биринчи чакирувда барча параметрлар мос аргументлар орқали қийматларини қабул қилади, иккинчи ҳолда I параметри 2 қийматини, байроқ параметри false қийматини ва Blg ўзгарувчиси келишув бўйича '\n' қийматини қабул қилади.

Келишув бўйича қиймат беришнинг битта шarti бор - параметрлар рўйхатида келишув бўйича қиймат берилган параметрлардан кейинги параметрлар ҳам келишув бўйича қийматга эга бўлишлари шart. Юқоридаги мисолда I параметри келишув бўйича қиймат қабул қилинган ҳолда, Байроқ ёки Blg параметрлари қийматсиз бўлиши мумкин эмас. Мисол тариқасида берилган сонни кўрсатилган аниқликда чоп этувчи дастурни кўрайлик. Қўйилган масалани ечишда сонни даражага ошириш функцияси - pow() ва сузувчи нуқтали узун сондан модул олиш fabsl() функциясидан фойдаланилади. Бу функциялар прототипи «cmath» сарлавҳа файлида жойлашган:

```
#include <iostream>  
  
using namespace std;  
  
#include <cmath>
```

```

void Chop_qilish(double Numb, double Aniqlik=1,
    bool Bayroq=true);

int main()
{
    double Mpi=-3.141592654;
    Chop_qilish(Mpi, 4, false);
    Chop_qilish(Mpi, 2);
    Chop_qilish(Mpi);
    return 0;
}

void Chop_qilish(double Numb,double Aniqlik=1,
    bool Bayroq = true)
{
    if(!Bayroq)Numb=fabsl(Numb);
    Numb=(int)(Numb*pow(10,Aniqlik));
    Numb=Numb/pow(10,Aniqlik);
    cout<<Numb<<'\n';
}

```

Дастурда сонни турли аниқликда (Аниқлик параметри қиймати орқали) чоп этиш учун ҳар хил вариантларда Chop\_qilish( ) функцияси чақирилган. Дастур ишлаши натижасида экранда куйидаги сонлар чоп этилади:

```

3.1415
-3.14
-3.1

```

Параметрнинг келишув бўйича бериладиган қиймати ўзгармас, глобал ўзгарувчи ёки қандайдир функция томонидан қайтарилган қиймат бўлиши мумкин.

### 13.5. Кўриниш соҳаси. Локал ва глобал ўзгарувчилар. :: амали

Ўзгарувчилар функция танасида ёки ундан ташқарида эълон қилиниши мумкин. Функция ичида эълон қилинган ўзгарувчиларга *локал ўзгарувчилар* дейилади. Бундай ўзгарувчилар хотирадаги дастур стекида жойлашади ва фақат ўзи эълон қилинган функция танасида амал қилади. Бошқарув асосий функцияга қайтиши билан локал ўзгарувчилар учун ажратилган хотира бўшатилади (ўчирилади).

Ҳар бир ўзгарувчи ўзининг амал қилиш соҳаси ва яшаш вақти хусусиятлари билан характерланади.

Ўзгарувчи *амал қилиш соҳаси* деганда ўзгарувчини ишлатиш мумкин бўлган дастур соҳаси (қисми) тушунилади. Бу тушунча билан ўзгарувчининг *кўриниш соҳаси* узвий боғланган. Ўзгарувчи амал қилиш соҳасидан чиққанда кўринмай қолади. Иккинчи томондан, ўзгарувчи амал қилиш соҳасида бўлиши, лекин кўринмаслиги мумкин. Бунда кўриниш соҳасига рухсат бериш амали «::» ёрдамида кўринмас ўзгарувчига мурожат қилиш мумкин бўлади.

Ўзгарувчининг *яшаш вақти* деб, у мавжуд бўлган дастур бўлагининг бажарилишига кетган вақт интервалига айтилади.

Локал ўзгарувчилар ўзлари эълон қилинган функция ёки блок чегарасида кўриниш соҳасига эга. Блокдаги ички блокларда худди шу номдаги ўзгарувчи эълон қилинган бўлса, ички блокларда бу локал ўзгарувчи ҳам амал қилмай қолади. Локал ўзгарувчи яшаш вақти - блок ёки функцияни бажариш вақти билан аниқланади. Бу ҳол шуни англатадики, турли функцияларда бири-бирига умуман боғлиқ бўлмаган бир хил номдаги локал ўзгарувчиларни ишлатиш мумкин.

Қуйидаги дастурда `main()` ва `sum()` функцияларида бир хил номдаги ўзгарувчиларни ишлатиш кўрсатилган. Дастурда иккита соннинг йиғиндиси ҳисобланади ва чоп этилади:

```
#include <iostream>
using namespace std;
// funksiya prototipi
int sum(int a, int b);
```

```

int main()
{
    // lokal o'zgaruvchilar
    int x=r;
    int y=4;
    cout << sum(x, y);
    return 0;
}
int sum(int a,int b)
{
    // lokal o'zgaruvchi
    int x=a+b;
    return x;
}

```

Глобал ўзгарувчилар дастур матнида функция аниқланишидан ташқарида эълон қилинади ва эълон қилинган жойидан бошлаб дастур охиригача амал қилади.

```

#include <iostream>
using namespace std;
int f1();
int f2();
int main()
{
    cout<<f1()<<" "<<f2()<<endl;
    return 0;
}
int f1()

```

```

{
return x; // kompilyatsiya xatosi ro'y beradi
}

int x=10; // global o'zgaruvchi e'loni

int f2() { return x*x; }

```

Юқорида келтирилган дастурда компиляция хатоси рўй беради, чунки f1() функция учун x ўзгарувчиси номаълум ҳисобланади.

Дастур матнида глобал ўзгарувчиларни улар эълонидан кейин ёзилган ихтиёрий функцияда ишлатиш мумкин. Шу сабабли, глобал ўзгарувчилар дастур матнининг бошида ёзилади. Функция ичидан глобал ўзгарувчига мурожат қилиш учун функцияда унинг номи билан мос тушадиган локал ўзгарувчилар бўлмаслиги керак. Агар глобал ўзгарувчи эълонида унга бошланғич қиймат берилмаган бўлса, уларнинг қиймати 0 ҳисобланади. Глобал ўзгарувчининг амал қилиш соҳаси унинг кўриниш соҳаси билан устма-уст тушади.

Шуни қайд этиш керакки, тажрибали дастур тузувчилар имкон қадар глобал ўзгарувчиларни ишлатмасликка ҳаракат қилишади, чунки бундай ўзгарувчилар қийматини дастурнинг ихтиёрий жойидан ўзгартириш хавфи мавжудлиги сабабли дастур ишлашида мазмунан хатолар юзага келиши мумкин. Бу фикримизни тасдиқловчи дастурни кўрайлик.

```

#include <iostream>

using namespace std;

// global o'zgaruvchi e'loni
int test = 100;

void Chop_qilish(void);

int main()
{
//lokal o'zgaruvchi e'loni
int test=10;

```

```

//global o'zgaruvchi chop qilish funksiyasini chaqirish
Chop_qilish();
cout << "Lokal o'zgaruvchi: " << test << '\n';
return 0;
}
void Chop_qilish(void)
{
cout<< "Global o'zgaruvchi: " << test << '\n';
}

```

Дастур бошида тест глобал ўзгарувчиси 100 қиймати билан эълон қилинади. Кейинчалик, main() функциясида тест номи билан локал ўзгарувчиси 10 қиймати билан эълон қилинади. Дастурда, Chop\_qilish() функциясига мурожаат қилинганида, асосий функция танасидан вақтинча чиқилади ва натижада main() функциясида эълон қилинган барча локал ўзгарувчиларга мурожаат қилиш мумкин бўлмай қолади. Шу сабабли Chop\_qilish() функциясида глобал тест ўзгарувчисининг қийматини чоп этилади. Асосий дастурга қайтилгандан кейин, main() функциясидаги локал тест ўзгарувчиси глобал тест ўзгарувчисини «беркитади» ва локал тест ўзгарувчини қиймати чоп этилади. Дастур ишлаши натижасида экранга куйидаги натижалар чоп этилади:

```
Global o'zgaruvchi: 100
```

```
Lokal o'zgaruvchi: 10
```

:: амали.

Юқорида қайд қилингандек, локал ўзгарувчи эълони худди шу номдаги глобал ўзгарувчини «беркитади» ва бу жойдан глобал ўзгарувчига мурожат қилиш имкони бўлмай қолади. С++ тилида бундай ҳолатларда ҳам глобал ўзгарувчига мурожат қилиш имко-нияти сақланиб қолинган. Бунинг учун «кўриниш соҳасига рухсат бериш» амалидан фойдаланиш мумкин ва ўзгарувчи олдида иккита нуқта - «::» қўйиш зарур бўлади. Мисол тариқасида куйидаги програмани келтирамиз:



```

#include <iostream>
using namespace std;
//global o'zgaruvchi e'loni
int uzg=5;
int main()
{
//lokal o'zgaruvchi e'loni
int uzg=70;
//lokal o'zgaruvchini chop etish
cout << uzg << '\n';
//global o'zgaruvchini chop etish
cout << ::uzg << '\n';
return 0;
}

```

Дастур ишлаши натижасида экранга олдин 70 ва кейин 5 сонлари чоп этилади.

### **13.6. Жойлаштириладиган (inline) функциялар**

Компилятор ишлаши натижасида ҳар бир функция машина коди кўринишида бўлади. Агар дастурда функцияни чақириш кўрсатмаси бўлса, шу жойда функцияни адреси бўйича чақиришнинг машина коди шаклланади. Одатда функцияни чақириш протсессор томонидан қўшимча вақт ва хотира ресурсларини талаб қилади. Шу сабабли, агар чақириладиган функция ҳажми унчалик катта бўлмаган ҳолларда, компиляторга функцияни чақириш коди ўрнига функция танасини ўзини жойлаштиришга кўрсатма бериш мумкин. Бу иш функция прототипини инлине калит сўзи билан эълон қилиш орқали амалга оширилади. Натижада ҳажми ошган, лекин нисбатан тез бажариладиган дастур коди юзага келади.

Функция коди жойлаштириладиган дастурга мисол.

```

#include <iostream>
using namespace std;
inline int Summa(int, int);
int main()
{
int a=2,b=6,c=3;
char yangi_qator = '\n';
cout << Summa(a,b) << yangi_qator;
cout << Summa(a,c) << yangi_qator;
cout << Summa(b,c) << yangi_qator;
return 0;
}
int Summa(int x, int y)
{
return x+y;
}

```

Келтирилган дастур кодини ҳосил қилишда Summa() функцияси чақирилган жойларга унинг танасидаги буйруқлар жойлаштирилади.

### 13.7. Қайта юкланувчи функциялар

Айрим алгоритмлар берилганларнинг ҳар хил турдаги қийматлари учун қўлланиши мумкин. Масалан, иккита соннинг максимумини топиш алгоритмида бу сонлар бутун ёки ҳақиқий турда бўлиши мумкин. Бундай ҳолларда бу алгоритмлар амалга оширилган функциялар номлари бир хил бўлгани маъқул. Бир нечта функцияни бир хил номлаш, лекин ҳар хил турдаги параметрлар билан ишлатиш *функцияни қайта юклаш* дейилади.

Компилятор параметрлар турига ва сонига қараб мос функцияни чақиради. Бундай амални «*ҳал қилиш амали*» дейилади ва унинг мақсади параметрларга кўра айнан (нисбатан) тўғри келадиган функцияни чақиритишдир. Агар бундай функция топилмаса компилятор хатолик ҳақида

хабар беради. Функцияни аниқлашда функция қайтарувчи қиймат турининг аҳамияти йўқ. Мисол:

```
#include <iostream>
using namespace std;
int max(int,int);
char max(char,char);
float max(float,float)
int max(int,int,int);
void main()
{
    int a,int b,char c,char d,int k,float x,y;
    cin>>a>>b>>k>>c>>d>>x>>y;
    cout<<max(a,b)<<max(c,d)<<max(a,b,k)<<max(x,y);
}
int max(int i,int j){return (i>j)?i:j;}
char max(char s1,char s2){return (s1>s2)?s1:s2;}
float max(float x,float y){return (x>y)?x:y;}
int max(int i,int j,int k)
{
    return (i>j)?(i>k? i:k;):((j>k)?j:k;);
}
```

Агар функция чақирилишида аргумент тури унинг прототипидаги худди шу ўриндаги параметр турига мос келмаса, компилятор уни параметр турига келтирилишга ҳаракат қилади - bool ва char турларини int турига, float турини double турига ва int турини double турига ўтказишга.

Қайта юкланувчи функциялардан фойдаланишда қуйидаги қоидаларга риоя қилиш керак:

- қайта юкланувчи функциялар битта кўриниш соҳасида бўлиши керак;

- қайта юкланувчи функцияларда келишув бўйича параметрлар ишлатилса, бундай параметрлар барча қайта юкланувчи функцияларда ҳам ишлатилиши ва улар бир хил қийматга эга бўлиш керак;

- агар функциялар параметрларининг тури фақат «const» ва ‘&’ белгилари билан фарқ қиладиган бўлса, бу функциялар қайта юкланмайди.

### **Адабиётлар рўйхати:**

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.- М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### **Мустақил таълим мавзулари:**

1. Функциялар ёрдамида бир нечта қиймат қайтариш.
2. Глобал ўзгарувчи ва константалар.
3. Глобал ўзгарувчилардан фойдаланиш.
4. Ташқи кутубхона файллари билан ишлаш.

### **Кейслар банки**

**Кейс 1.** Функциялар билан ишловчи дастур тузилди. Функциянинг вазифаси уч хонали соннинг жуфт рақамларини чоп этиш. Функцияни ихтиёрий узунликдаги (тур ўлчамига сиғадиган) сон учун ишлайдиган қилиб ўзгартириш керак.

### **Кейсни бажариш босқчилари ва топшириқлар:**

- Кейсдаги муаммони ҳал қилиш имконини берувчи бажарилиши керак бўлган ишлар кетма-кетлигини белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

### Глоссарий:

Термин	Terminology	Ўзбек тилидаги шархи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади.
Функция	Function	программа алоҳида бўлаги, асосий қисм томонидан чақириб ишлатилади
Параметр	Parametr	функция ишлаши учун керак берилганлар
аргумент	Argument	функцияга параметрига жўнатиладиган қиймат
сарлавҳа файли	Header file	функциялар эълони ёзилган файл
cout		экранга чиқариш оқими.
cin		экрандан киритиш оқими.
if		шарт оператори
Амал қилиш соҳаси		ўзгарувчини ишлатиш мумкин бўлган дастур соҳаси
Ўзгарувчи	Variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
Идентификатор	Identifier	катта ва кичик латин ҳарфлари, рақамлар ва таг чизиқ ('_') белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги.
else		шарт ёлфонлигини аниқловчи оператор

### Назорат саволлари:

1. Функция бу...?
2. Функциялар модуллар деб ҳам аталиши мумкунми?
3. С++ тилида функция чақирилганда айрим аргументларни тушириб қолдириш мумкунми ва бунга қандай еришиш мумкун?
4. Локал ўзгарувчилар ўзлари эълон қилинган функция ёки блок чегарасида кўриниш соҳасига эга буладими?
5. Функциялар қандай турларга бўлинади?
6. Келишув бўйича қиймат беришнинг нечта шарти бор?
7. С++ тилидаги ҳар қандай дастурда қайси функция бош функция болиши керак?
8. Функциялар қандай кўринишда бўлади?
9. Функция қандай аникланади?
10. Компилятор ишлаши натижасида ҳар бир функция қандай кўринишида бўлади?
11. Локал ўзгарувчи яшаш вақти қандай аникланади?
12. Include оператори қандай функцияни бажаради?
13. Cout оператори қандай функцияни бажаради?
14. Cin оператори қандай функцияни бажаради?
15. If оператори қандай функцияни бажаради?
16. Else оператори қандай функцияни бажаради?
17. Айрим алгоритмлар берилганларнинг қандай турдаги қийматлари учун қўлланиши мумкин?
18. Қайта юкланувчи функциялардан фойдаланишда қандай қоидаларга риоя қилиш керак?
19. Inline оператори қандай функцияни бажаради
20. Float оператори қандай функцияни бажаради

### Тест саволлари:

1. Функция эълони тўғри кўрсатилган жавобни топинг.
  - a) void F(int t=1)
  - b) void F(int t=1, int s)
  - c) void F(int t=int q)
  - d) void F(int t=double)

2. Функция эълони тўғри кўрсатилган жавобни топинг?

- a) void F(int t, double d)
- b) void F(int t=1, double s)
- c) void F(int t, double d=t)
- d) void F(int t=doubled)c

3. Функция эълони тўғри кўрсатилган жавобни топинг

- a) void F(char c=99, int t=99)
- b) void F(char c=99, int t)
- c) void F(char c=t, int t)
- d) void F(char c, int c)

4. Программа ишлаши натижасида экранга нима чиқади?

```
double SUM(double *a, int n)
{
double t=0;
t=t+a[0];
for (int i=1;i<n;i++)t=t+a[i]/a[i-1];
return t;
}
void main()
{
double a[]={7,1,3,5,3},c;
c=SUM(a,3);
cout << c;
}
```

- a) 10
- b) 20
- c) 30
- d) 50

5. Программа ишлаши натижасида экранга нима чиқади?

```
int F(int n, int k=3, int t=13)
{
n=5;
for (int i=0; i<=k;i++)
    t=t+n;
return n;
}
void main()
{
int a[]={3,6,5,7,9,1}, n=a[0], k=a[1], t=a[3];
for (int i=0;i<6;i++)
if (F(a[i])>a[i]) t=t+F(a[i],a[2],a[0]);
cout << t;
}
```

- a) 17
- b) 18
- c) 19
- d) 20



6. Програма ишлаши натижасида экранга нима чиқади?

```
int F(double &a)
{
a=13;
int b=a+41;
}
void main()
{
double b=15;
F(b);

        cout << b;

    }
```

a) 13

b) 54

c) 41

d) 22

7. Програма ишлаши натижасида экранга нима чиқади?

```
int F(int n, int k, int t)
{
for (int i=0; t; )
    {t--;n=n+k;}
return n;
}
void main()
{
int a[]={2,5,2,6,7}, n=2, k=1, t=3;
```

```
for (int i=0;i<3;i++)
if (2*F(a[i], a[i], a[0])>t) t=t+F(a[i],t,a[0]);
    cout << t;
}
```

a) 38

b) 41

c) 25

d) 57

8. Программа ишлаши натижасида экранга нима чиқади?

```
int F(int a)
{
a=5;
return a;
}
void main()
{
int a=7;
F(a)+F(a+5);
    cout << a;
}
```

A)7

b) 8

c) 5

d) 75

9. Программа ишлаши натижасида экранга нима чиқади?

```
int F(int n, int k, int t)
{
for (int i=0; t; )
    {t--;n=n+k;}
return n;
}

void main()
{
int a[]={3,3,3,6,7}, n=2, k=1, t=3;
for (int i=0;i<1;i++)
if (2*F(a[i], a[i], a[0])>t) t=t+F(a[i],t,a[0]);
cout << t;
}
```

- a) 15
- b) 25
- c) 3
- d) 21

10. Ушбу функция қандай қиймат қайтаради?

```
int funksiya(int n)
{
int k = 0;
while(n > 0)
{
k++;
n /= 10;
}
```

```
    }  
    return k;  
}
```

- a) n ниинг рақамлари сонини
- b) n нинг рақамлари йиғиндисини
- c) Ҳар доим нол қиймат қайтаради
- d) Қиймат қайтармайди

11. Ушбу функция қандай қиймат қайтаради?

```
int funksiya(int n)  
{ int s=0;  
    while(n > 0)  
    {  
        int a=n%10;  
        s=s+a;  
n /= 10;  
    }  
    return s;  
}
```

- a) n ниинг рақамлари йиғиндисини
- b) n нинг рақамлари йиғиндисини
- c) Ҳар доим нол қиймат қайтаради
- d) n ниинг рақамлари сонини

12. Ушбу функция қандай қиймат қайтаради?

```
int funksiya(int n)
{
    int s=0;
    while(n > 0)
    {
        int a=n%10;
        s=s*10+a;
        n=n/10;
    }
    return s;
}
```

- a) n рақамининг тескарисини
- b) n нинг рақамлари йиғиндисини
- c) n ниинг рақамлари сонини
- d) Қиймат қайтармайди

13. Программа ишлаши натижасида экранга нима чиқади?

```
int F(int n, int k=15, int t=12)
{
    return n*k/t;
}

void main()
{
    int n=7, a[]={5,2,-3,2,4}, c=2, k=2;
    for (int i=0;i<5;i++)
        if (2*F(a[i])-F(n-a[i],c)>F(a[i], a[i], a[i/2]))
            c=c+F(a[i], n);
    cout << c;
}
```

- a) 7
- b) 8
- c) 15
- d) 12

14. Программа ишлаши натижасида экранга нима чиқади?

```
int F(int n, int k=5)
{
k=3;
return n-k;
}
void main()
{
int n=7, a[]={5,2,3,2,4}, c=2, k=2;
for (int i=0;i<5;i++)
if (2*F(a[i])-F(n-a[i],c)>0) c=c+F(a[i], n);
cout << c;
}
```

- a) 5
- b) 6
- c) 7
- d) 8

15. Программа ишлаши натижасида экранга нима чиқади?

```
int F(int n, int k=2, int t=6)
{
for (;t;)t=t-2;
```

```

return (n+t)/k;
}
void main()
{
int n=8, a[]={5,2,-3,2,4}, c=2, k=2;
for (int i=0;i<5;i++)
if (2*F(a[i])-F(c-a[i],n)>F(a[i], a[i], n)) c=c+F(a[i]);
cout << c;
}

```

- a) 8
- b) 9
- c) 10
- d) 11

16. Програма ишлаши натижасида экранга нима чиқади?

```

int F(int a, int b)
{
return (a+b);
}
void main()
{
int a=3, b=4, c=7;
cout << F(b,c);
}

```

- a) 11
- b) 7
- c) 10
- d) 34

17. Программа ишлаши натижасида экранга нима чиқади?

```
int F(int a, int b)
{
return (a+b);
}

void main()
{
int a=3, b=4, c=7;
cout << F(a+b, c+7);
}
```

- a) 11
- b) 7
- c) 10
- d) 34

18. Программа ишлаши натижасида экранга нима чиқади?

```
bool F(int a, int b)
{
return (a+b);
}

void main()
{
int a=3, b=4;
cout << F(a,b);
}
```

- a) 1
- b) 7
- c) 0
- d) 34



19. Программа ишлаши натижасида экранга нима чиқади?

```
bool F(int a, int b)
{
return a?b:0;
}

void main()
{
int a=3, b=4;
cout << F(b,c);
}
```

- a) 1
- b) 4
- c) 0
- d) 7

20. Программа ишлаши натижасида экранга нима чиқади?

```
int F(int a, int b)
{
return a?b:0;
}

void main()
cout << F(b,c);
}
```

- a) 4
- b) 1
- c) 0
- d) 7

## Маъруза 15. Рекурсив функциялар

### Режа:

1. Рекурсия тушунчаси.
2. «Ханой минораси» масаласи.
3. Фибоначчи сонларини топиш масаласи.

**Таянч атамалар:** *include, cout, cin, if, for, return, рекурсия, Функция, Параметр, аргумент, Ўзгарувчи, Идентификатор.*

### 15.1. Рекурсия тушунчаси

Одатда рекурсия математикада кенг қўлланилади. Чунки аксарият математик формулалар рекурсив аниқланади. Мисол тариқасида факториални ҳисоблаш формуласини

$$n! = \begin{cases} 1, & \text{агар } n = 0; \\ n * (n - 1)!, & \text{агар } n > 0, \end{cases}$$

ва соннинг бутун даражасини ҳисоблашни кўришимиз мумкин:

$$x^n = \begin{cases} 1, & \text{агар } n = 0; \\ x * x^{n-1}, & \text{агар } n > 0. \end{cases}$$

Кўриниб турибдики, навбатдаги қийматни ҳисоблаш учун функциянинг «олдинги қиймати» маълум бўлиши керак. С++ тилида рекурсия математикадаги рекурсияга ўхшаш.

Қуйидаги масала қаралсин: математикада манфий бўлмаган бутун сонларнинг факториалини аниқлаш қуйидаги формула ёрдамида амалга оширилади:

$$0! = 1 \quad (1)$$

$$n! = n \times (n-1)! \quad (2)$$

Агар  $n=3$  бўлса масала қуйидаги формулалар ёрдамида ишланади:

$$3! = 3 \times 2!$$

$$2! = 2 \times 1!$$

$$1! = 1 \times 0!$$

$$0! = 1$$

(1) формула тўғри ишланади, чунки ифоданинг ўнг қисмида факториални ҳисоблаш мавжуд эмас. (2) формулада эса ифоданинг ўнг қисмида яна факториални ҳисоблаш керак. (1) ва (2) формулалар рекурсив формулалар дейилади. (1) ифода асос ифода, (2) ифода умумий ифода дейилади.

Рекурсия деб функция танасида шу функциянинг ўзини чақиршига айтилади. Рекурсия учун қуйидаги аниқланишлар ўринли:

1. Ҳар бир рекурсив формула камида битта асос ифодага эга бўлиши керак.
2. Умумий ифода доим асос ифодага йўналтирилган бўлиши керак.
3. Асос ифода рекурсияни тўхтатиши керак.

Буни юқоридаги мисоллар учун тузилган функцияларда кўриш мумкин.

Факториал учун:

```
long F(int n)
{
    if (n == 0)
        return 1;
    else
        return n * F(n-1);
}
```

Берилган ҳақиқий  $x$  сонинг  $n$  - даражасини ҳисоблаш функцияси:

```
double Butun_Daraja(double x, int n)
{
    if (n == 0)
        return 1;
    else
        return x * Butun_Daraja(x, n-1);
}
```

Агар факториал функциясига  $n > 0$  қиймат берилса, қуйидаги ҳолат рўй беради: шарт операторининг элсе шохидаги қиймати ( $n$  қиймати) стекда эслаб қолинади. Ҳозирча қиймати номаълум  $n-1$  факториални ҳисоблаш учун шу функциянинг ўзи  $n-1$  қиймати билан билан чақирилади. Ўз навбатида, бу

қиймат ҳам эслаб қолинади (стекка жойланади) ва яна функция чақирилади ва ҳакоза. Функция  $n=0$  қиймат билан чақирилганда иф операторининг шарти ( $n == 0$ ) рост бўлади ва «return 1;» амали бажарилиб, айти шу чақириш бўйича 1 қиймати қайтарилади. Шундан кейин «тескари» жараён бошланади - стекда сақланган қийматлар кетма-кет олинади ва кўпайтирилади: охириги қиймат - аниқлангандан кейин (1), у ундан олдинги сақланган қийматга 1 қийматига кўпайтириб  $F(1)$  қиймати ҳисобланади, бу қиймат 2 қийматига кўпайтириш билан  $F(2)$  топилади ва ҳакоза. Жараён  $F(n)$  қийматини ҳисоблашгача «кўтарилиб» боради. Бу жараёни,  $n=4$  учун факториал ҳисоблаш схемасини қуйида кўриш мумкин:

↓ $F(4)=4 * F(3)$	↓ $F(4)=4 * F(3)$	↓ $F(4)=4 * F(3)$	↓ $F(4)=4 * F(3)$	↑ $F(4)=4 * 6$
↓ $F(3)=3 * F(2)$	↓ $F(3)=3 * F(2)$	↓ $F(3)=3 * F(2)$	↑ $F(3)=3 * 2$	
↓ $F(2)=2 * F(1)$	↓ $F(2)=2 * F(1)$	↑ $F(2)=2 * 1$		
↓ $F(1)=1 * F(0)$	↑ $F(1)=1 * 1$			
↑ $F(0)=1$				

4! ҳисоблаш схемаси

Рекурсив функцияларни тўғри амал қилиши учун рекурсив чақиришларнинг тўхташ шарти бўлиши керак. Акс ҳолда рекурсия тўхтамаслиги ва ўз навбатида функция иши тугамаслиги мумкин. Факториал ҳисоблашида рекурсив тушишларнинг тўхташ шарти функция параметри  $n=0$  бўлишидир (шарт операторининг рост шохи).

Рекурсия икки хил бўлади:

1. оддий - агар функция ўз танасида ўзини чақирса;
2. воситали - агар биринчи функция иккинчи функцияни чақирса, иккинчиси эса ўз навбатида биринчи функцияни чақирса.

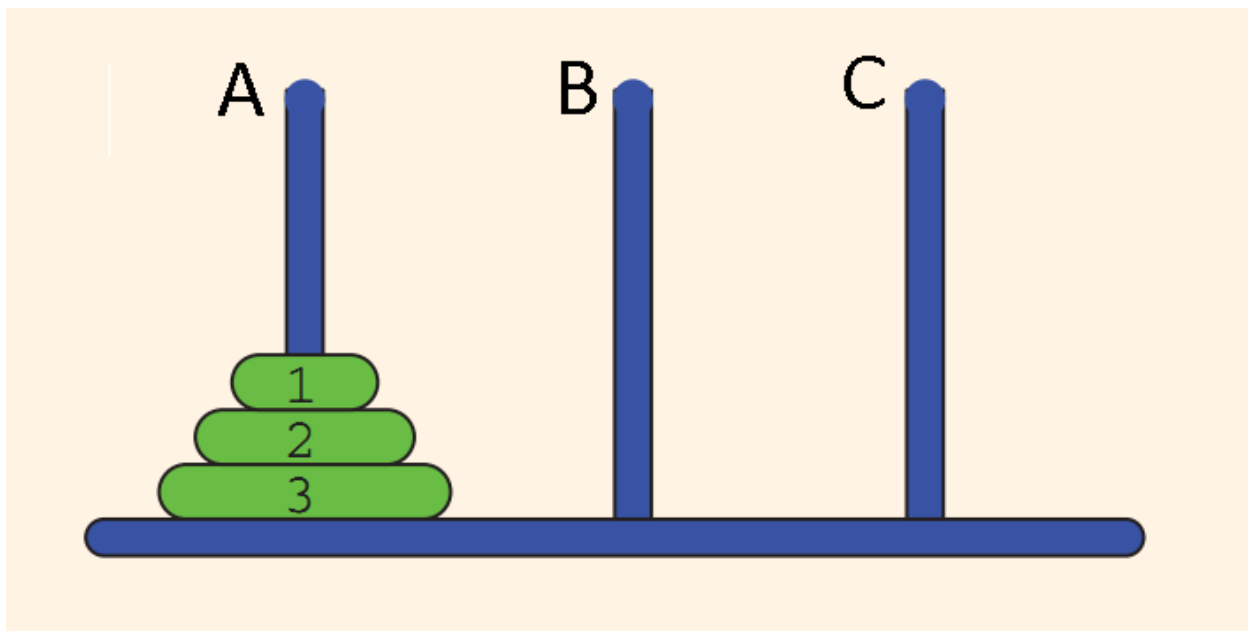
Ҳар бир рекурсив мурожаат қўшимча хотира талаб қилади - функцияларнинг локал объектлари (ўзгарувчилари) учун ҳар бир мурожаатда стекдан янгидан жой ажратилади. Масалан, рекурсив функцияга 100 марта мурожаат бўлса, жами 100 локал объектларнинг мажмуаси учун жой ажратилади. Айрим ҳолларда, яъни рекурсиялар сони етарлича катта бўлганда,

стек ўлчами чекланганлиги сабабли (реал режимда 64Кб ўлчамгача) у тўлиб кетиши мумкин. Бу ҳолатда дастур ўз ишини «Стек тўлиб кетди» хабари билан тўхтади.

### 15.2. «Ханой минораси» масаласи

Қуйида, рекурсия билан самарали ечиладиган «Ханой минораси» масаласини кўрайлик.

**Масала.** Учта А, В, С қозик ва n-та ҳар хил ўлчамли халқалар мавжуд. Халқаларни ўлчамлари ўсиш тартибида 1 дан n гача тартибланган. Бошда барча халқалар А қозикқа расмдагидек жойлаштирилган. А қозикдаги барча халқаларни С қозикқа, ёрдамчи В қозикдан фойдаланган ҳолда, қуйидаги қоидаларга амал қилган ҳолда ўтказиш талаб этилади: халқаларни биттадан кўчириш керак ва катта ўлчамли халқани кичик ўлчамли халқа устига қўйиш мумкин эмас.



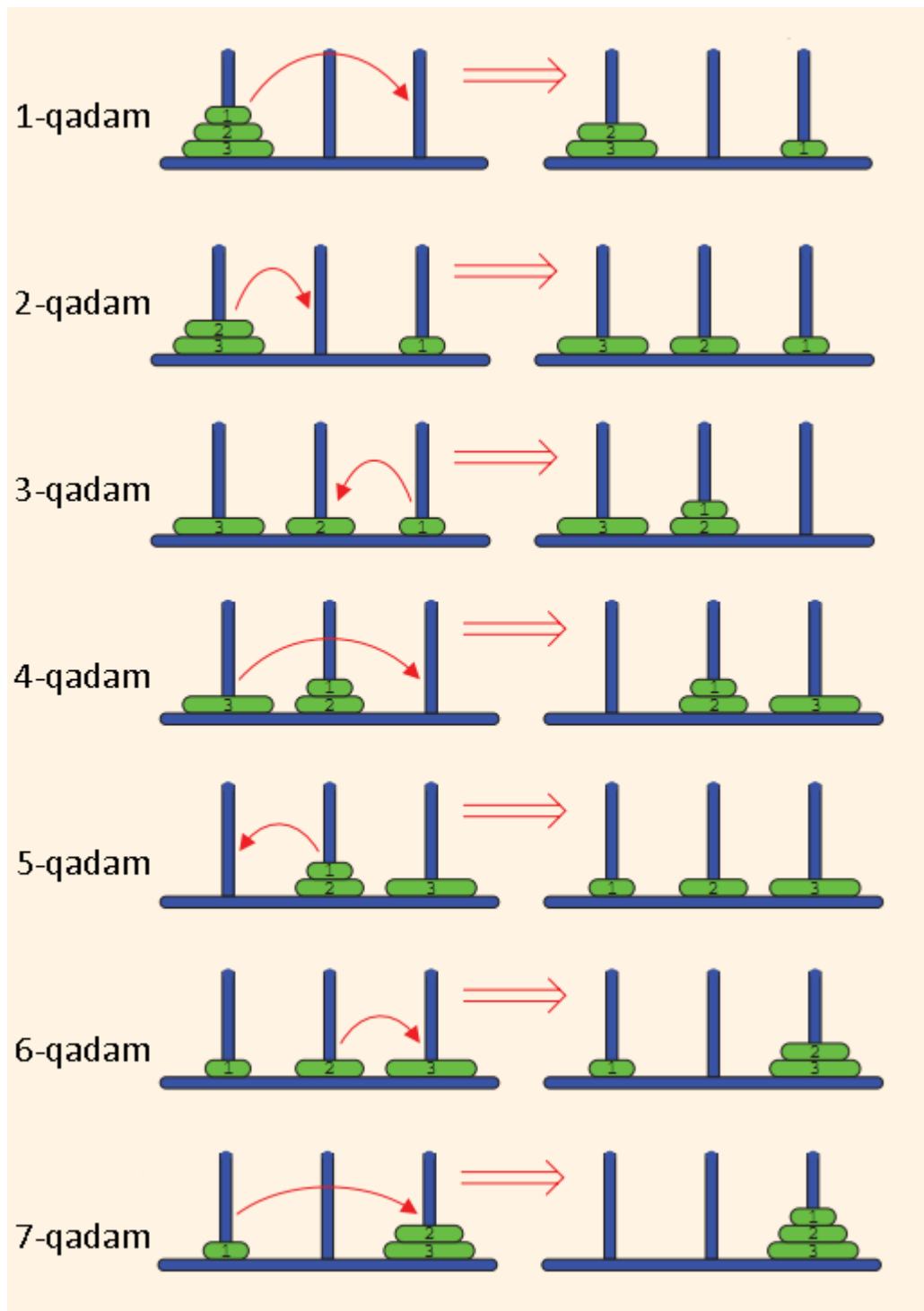
Ханой минораси масаласи

```
#include <iostream>
using namespace std;
void Hanoy(int n, char a = 'A', char b = 'C',
           char c = 'B')
```

```

{
if(n)
{
    Hanoy(n-1, a, c, b);
    cout << "Xalqa " << a << " dan " << b
        << " ga o'tkazilsin\n";
    Hanoy(n-1, c, b, a);
}
}
int main()
{
    unsigned int Xalqalar_Soni;
    cout << "Hanoy minorasi masalasi" << endl;
    cout << "Xalqalar sonini kiriting: ";
    cin >> Xalqalar_Soni;
    Hanoy(Xalqalar_Soni);
    return 0;
}

```



Халқалар сони 3 бўлганда ( $Xalqalar\_Soni=3$ ) дастур экранга халқаларни кўчириш бўйича амаллар кетма-кетлигини чоп этади:

- Xalqa A dan C ga o'tkazilsin
- Xalqa A dan B ga o'tkazilsin
- Xalqa C dan B ga o'tkazilsin
- Xalqa A dan C ga o'tkazilsin

Xalqa B dan A ga o'tkazilsin

Xalqa B dan C ga o'tkazilsin

Xalqa A dan C ga o'tkazilsin

Тахлил қилиб кўрилса, учта халқани А қозикдаги барча халқаларни С қозикқа ўтказиш учун  $2^3-1=7$  та жараён бажарилди. Халқалар сони 64та бўлганда бу жараёнлар сони  $2^{64}-1$  га тенг бўлади.

$$2^{10} = 1024 \approx 1000 = 10^3$$

Бундан келиб чиқадики:

$$2^{64} = 2^4 \times 2^{60} \approx 2^4 \times 10^{18} = 1.6 \times 10^{19}$$

Бир йилдаги секундлар сони  $3.2 \times 10^7$  га тенг. Бир дона дискни бир қозикдан бошқасига олиб ўтиш учун бир секунд сарфланади деб ҳисобланса қуйидагига келиш мумкин:

$$1.6 \times 10^{19} = 5 \times 3.2 \times 10^{18} = (3.2 \times 10^7) \times (5 \times 10^{11})$$

Барча 64та дискни А қозикдан С қозикқа олиб ўтиш учун  $(5 \times 10^{11})$  йил керак бўлади. Компьютер бир секундда бир миллиард ( $10^9$ ) операция бажара олади деб ҳисобланса, бир йилда қуйидагича операция бажара олади:

$$(3.2 \times 10^7) \times 10^9 = 3.2 \times 10^{16}$$

64та дискни А қозикдан С қозикқа олиб ўтиш учун компьютерга қуйидагича миқдорда вақт керак бўлади:

$$2^{64} \approx 1.6 \times 10^{19} = 1.6 \times 10^{16} \times 10^3 = (3.2 \times 10^{16}) \times 500$$

Яъни, 500 йил вақт керак бўлади.

### 15.3. Фибоначчи сонларини топиш масаласи

Фибоначчи сонларини топишда ҳар бир ҳад ўзидан олдинги икки ҳад йиғиндисига тенг. Биринчи ва иккинчи хади олдиндан маълум бўлади. Рекурсияни қўллаган ҳолда фибоначчи сонларини топиш формуласини қуйидагича ёзиш мумкин:

$$rFibNum(a,b,n) = \begin{cases} a & \text{àãàð } n = 1 \\ b & \text{àãàð } n = 2 \\ rFibNum(a,b,n-1) + rFibNum(a,b,n-2) & \text{àãàð } n > 2 \end{cases}$$

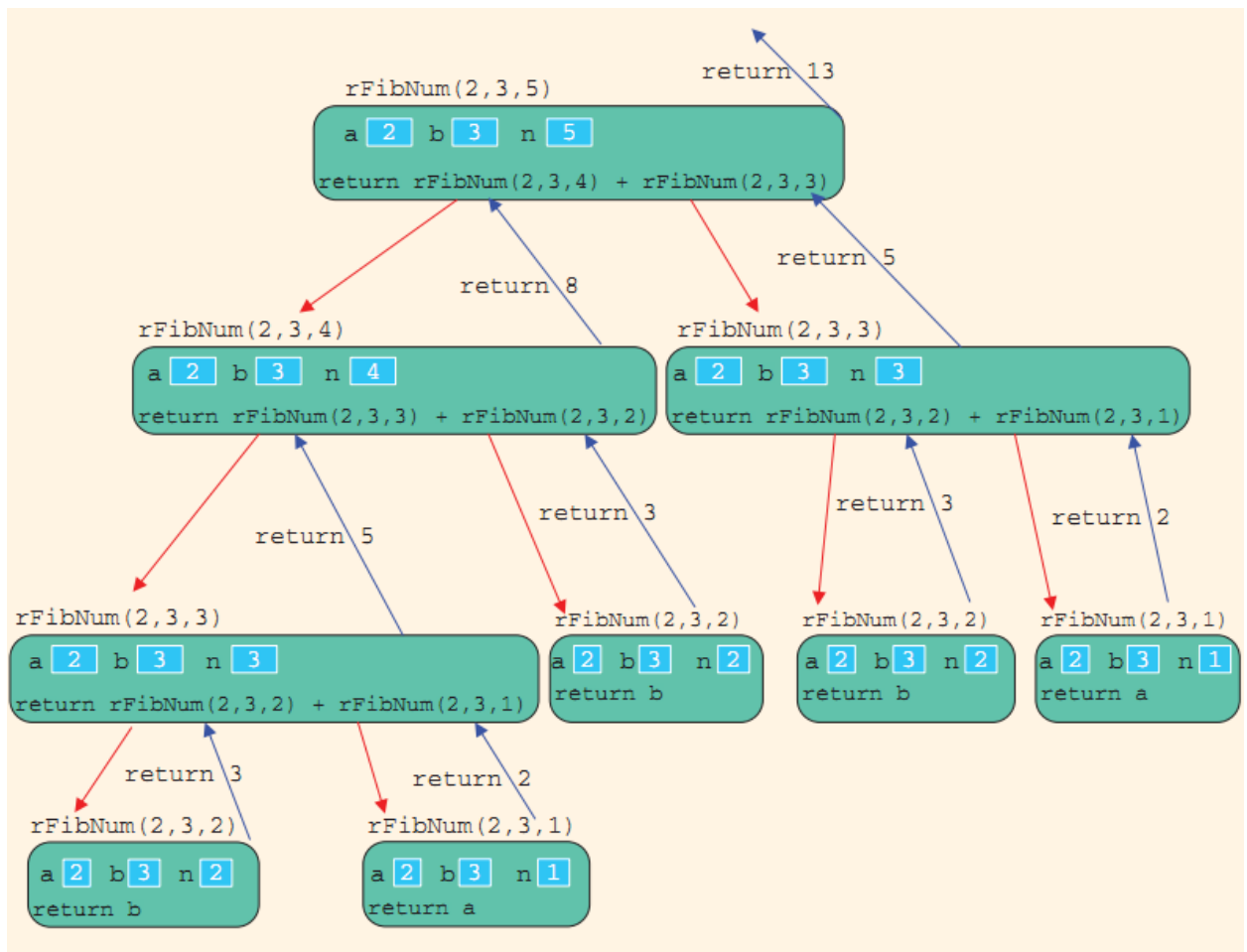
```
#include <iostream>
using namespace std;
```



```

int rFibNum(int a, int b, int n);
int main()
{
int firstFibNum;
int secondFibNum;
int n;
cout << "Birinchi fibonachchi sonini kiriting: ";
cin >> firstFibNum;
cout << endl;
cout << "Ikkinchi fibonachchi sonini kiriting: ";
cin >> secondFibNum;
cout << endl;
cout << "Qidirilayotgan fibonachchi soni o'rnini
kiriting: ";
cin >> n;
cout << endl;
cout << n << " - o'rindagi Fibonachchi soni: "
    << rFibNum(firstFibNum, secondFibNum, n) <<endl;
return 0;
}
int rFibNum(int a, int b, int n)
{
if (n == 1)
    return a;
else if (n == 2)
    return b;
else
    return rFibNum(a, b, n - 1)
        + rFibNum(a, b, n - 2);
}

```



Функцияни чақириш учун қуйидаги кодни ёзиш керак:

```
rFibNum(2, 3, 5) << endl;
```

Рекурсия чиройли, ихчам кўрингани билан хотирани тежаш ва ҳисоблаш вақтини қисқартириш нуқтаи-назаридан уни имкон қадар итератив ҳисоблаш билан алмаштирилгани маъқул. Масалан,  $x$  ҳақиқий сонининг  $n$ -даражасини ҳисоблашнинг қуйидаги ечим варианти нисбатан кам ресурс талаб қилади ( $n$  – бутун ишорасиз сон):

```
double Butun_Daraja(double x, int n)
{
    double p=1;
    for(int i=1; i<=n; i++)p*=x;
    return p;
}
```

Иккинчи томондан, шундай масалалар борки, уларни ечишда рекурсия жуда самарали, ҳаттоки ягона усулдир. Хусусан, грамматик таҳлил масалаларида рекурсия жуда ҳам ўнғай ҳисобланди.

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим

1. Воситали рекурсия.
2. Воситасиз рекурсия.
3. ЭКУБ масаласи.

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
cout		экранга чиқариш оқими
cin		экрандан киритиш оқими
if		шарт оператори
for		такрорлаш оператори
return		функция қийматини қайтарувчи оператор
рекурсия	Recursion	функция танасида функцияни чақириш

Функция	Function	программинг алоҳида бўлаги, асосий қисм томонидан чақириб ишлатилади
Параметр	Parametr	функция ишлаши учун керак берилганлар
аргумент	Argument	функцияга параметрига жўнатиладиган қиймат
Ўзгарувчи	variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
Идентификатор	identidier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизиқ ('_') белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги

### Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур иккита соннинг ЭКУБини рекурсия ёрдамида топиш учун мўлжалланган. Лекин дастур ишлаши натижасида киритилган икки соннинг ЭКУБи доим нол чиқмоқда. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

## Назорат саволлари

1. Қийматни ҳисоблаш учун функциянинг «олдинги қиймати» маълум бўлиши керакми?
2. Математикада манфий бўлмаган бутун сонларнинг факториалини аниқлаш қайси формула ёрдамида амалга оширилади?
3. Рекурсия деб нимага айтилади?
4. Рекурсия учун қандай аниқланишлар ўринли?
5. Агар факториал функциясига  $n > 0$  қиймат берилса, қандай ҳолат рўй беради?
6. Рекурсив функцияларни тўғри амал қилиши учун қандай чақиришларнинг тўхташ шарти бўлиши керак?
7. Ҳар бир рекурсив мурожаат кўшимча хотира талаб қиладими?
8. Рекурсия чиройли, ихчам кўрингани билан хотирани тежаш ва ҳисоблаш вақтини қисқартириш нуқтаи-назаридан уни имкон қадар итератив ҳисоблаш билан алмаштирилгани маъқулми?
9. Рекурсия қандай тўхтатилади?
10. Ҳар бир рекурсив формула нечта ифодага эга бўлиши керак?

## Тест саволлари

1. Қуйидаги рекурсив функция бажарилиши натижасида экранга нима чиқади.

```
long F(int n)
{ if(!n) return 1;
  else return n*F(n-1);
}
int main()
{long s; int n=5;
  s=F(n);
cout<<"s="<<s;
}
```

- a) 120
- b) 130
- c) 24
- d) 620

2. Қуйидаги рекурсив функция бажарилиши натижасида экранга нима чиқади.

```
long F(int n)
{ if(!n) return 1;
  else return n+F(n-1);
} int main()
{ long s; int n=10;
  s=F(n);
cout<<"s="<<s;
}
```

- a) 56
- b) 45
- c) 46
- d) 55

3. Қуйидаги рекурсив функция бажарилиши натижасида экранга нима чиқади.

```
double rec(double x, int n)
{ if(!n) return 1;
  else return x*rec(x,n-1);
} int main()
{ long s; int x=4;
  s=rec(x,4);
cout<<"s="<<s;
}
```

- a) 256
- b) 64
- c) 16
- d) 4

4. Қуйидаги рекурсив функция бажарилиши натижасида экранга нима чиқади.

```
double rec(double x, int n)
{ if(!n) return 1;
  else return x*rec(x,n-1); }
int main()
{ long s; int x=4;
```

```
s=rec(1,4);  
cout<<"s="<<s;  
}
```

- a) 1
- b) 4
- c) 64
- d) 256

5. Ушбу функция нима жавоб қайтаради?

```
int funksiya(int n)  
{  
    int k = 0;  
    while(n > 0)  
    {  
        k++;  
        n /= 10;  
    }  
    return k;  
}
```

- a) Бутун сон қайтаради
- b) Ҳақиқий сон қайтаради
- c) Мантиқий қиймат қайтаради
- d) Қиймат қайтармайди

6. Қуйидаги рекурсив функция бажарилиши натижасида экранга нима чиқади.

```
long F(int n)  
{  
    if(n==1) return 1;  
    else return n+F(n-1);  
}  
int main()  
{long s; int n=10;  
    s=F(n);  
    cout<<"s="<<s;  
}
```

- a) 55
- b) 57
- c) 56
- d) 47

7. Қуйидаги рекурсив функция бажарилиши натижасида экранга нима чиқади.

```
float F(int n)
{  if(!n) return 0;
   else return 1./n+F(n-1);
}
int main()
{float s; int n=2;
  s=F(n);
cout<<"s="<<s;}
```

- a) 1.5
- b) 2.5
- c) 0.5
- d) 1

8. Қуйидаги рекурсив функция бажарилиши натижасида экранга нима чиқади.

```
float F(int n)
{  if(!n) return 1;
   else return 1./n*F(n-1);
}
int main()
{float s; int n=2;
  s=F(n);
cout<<"s="<<s;}
```

- a) 0.5
- b) 1.5
- c) 1
- d) 2



9. Қуйидаги рекурсив функция бажарилиши натижасида экранга нима чиқади.

```
float F(int n)
{  if(!n) return 0;
   else return n+F(n-1);
}
int main()
{float s; int n=4;
  s=F(n);
cout<<"s="<<s;}
```

- a) 10
- b) 15
- c) 20
- d) 5

10. Қуйидаги рекурсив функция бажарилиши натижасида экранга нима чиқади.

```
float F(int n)
{  if(!n) return 0;
   else return n+F(n-1);
}
int main()
{float s; int n=6;
  s=F(n);
cout<<"s="<<s;}
```

- a) 21
- b) 20
- c) 15
- d) 28

## Маъруза 16. Фойдаланувчи томонидан аниқланган берилганлар турлари

### Режа:

1. Санаб ўтилувчи тур.
2. Санаб ўтилувчи тур устида амаллар бажариш
3. Функциялар ва enum турлар
4. typedef билан ишлаш

**Таянч иборалар:** *include, typedef, enum, аргумент, ўзгармас, сарлавҳа файли, switch, cout, cin, if, else*

### 16.1. Санаб ўтилувчи тур

C++ да берилганларнинг оддий турлари учта категорияга бўлинади: бутун, нуқтали сузувчи ва санаб ўтилувчи тур. Берилганларнинг int тури -2 147 483 648 дан 2 147 483 647 гача бўлган бутун сонлардан ташкил топган ва улар устида арифметик амаллар бажарилади (+, -, \*, /, ва %). Модомики дастурнинг асосий мақсади берилганларни бошқариш экан, ихтиёрий программалаш тилида берилганларнинг турлари асосий тушунчларидан бири ҳисобланади.

Дастурлашда берилганларни янги турини аниқлашга тўғри келди. Берилганларнинг янги тури аниқлаш учун олдин берилганларни тури қандай берилганларни қабул қилади ва фойдали томонлари бор ва бу берилганлар устида фойдаланувчи қандай амаллар бажарилади. Бундай ҳолларда юз берадиган хатоликларни система бартараф этади.

Биз ҳозиргача фойдаланиб келаётган берилганларнинг int, bool, char ва double турлари берилганлар турларининг асосий турлари ҳисобланади. Ҳаттоки бу берилганларнинг турлари ихтиёрий турдаги масалаларни ечишда етарли, аммо бу турлар ўзига хос масалаларни ечишда бир хил эмас.

C++ тили фойдаланувчиларга программалашда имкониятларини ошириш учун берилганларнинг шахсий санаб ўтилувчи турини яратишга имкон беради.

Берилганларнинг санаб ўтилувчи турини аниқлаш қуйидаги қисмлардан иборат:

1. Берилганлар турнинг номи;
2. Берилганлар тури номининг қийматлари;
3. Берилганлар турнинг қийматлари устида қандай амаллар бажариш;

C++ берилганларнинг оддий санаб ўтилувчи турини ва улар қийматларини аниқлашга имкон беради аммо улар устида қандай амаллар бажаришни эмас.

Кўп миқдордаги, мантиқан боғланган ўзгармаслардан фойдаланилганда санаб ўтилувчи турдан фойдаланилгани маъқул. Санаб ўтилувчи ўзгармаслар `enum` калит сўзи билан аниқланади. Мазмуни бўйича бу ўзгармаслар оддий бутун сонлардир. Санаб ўтилувчи ўзгармаслар C++ стандарти бўйича бутун турдаги ўзгармаслар ҳисобланади. Ҳар бир ўзгармасга (сонга) мазмунли ном берилади ва бу идентификаторни программанинг бошқа жойларида номлаш учун ишлатилиши мумкин эмас.

Санаб ўтилувчи тур қуйидаги кўринишга эга:

```
enum <санаб ўтиладиган тур номи> { <ном1> =<қиймат1>,
                                   <ном 2> =<қиймат2>,
                                   ...
                                   <ном> =<қиймат н> };
```

Бу ерда, `enum` - калит сўз (инглизча `enumerate` - санамоқ);

<санаб ўтиладиган тур номи>- ўзгармаслар рўйхатининг номи; <ном и> - бутун

қийматли константаларнинг номлари; <қиймат и >- шарт бўлмаган инициализация қиймати (ифода).

Мисол учун ҳафта кунлари билан боғлиқ масала ечишда ҳафта кунларини душ (душанба), сеш (сешанба), чор (чоршанба), пайш (пайшанба), жума (жума), шанба (шанба), якш (якшанба) ўзгармасларини ишлатиш мумкин ва улар санаб ўтилувчи тур ёрдамида битта сатрда ёзилади:

```
enum Hafta {dush, sesh, chor, paysh, juma, shanba, yaksh};
```

Санаб ўтилувчи ўзгармаслар қуйидаги хоссага эга: агар ўзгармас қиймати кўрсатилмаган бўлса, у олдинги ўзгармас қийматидан биттага ортиқ бўлади. Келишув бўйича биринчи ўзгармас қиймати 0 бўлади.

Инициализация ёрдамида ўзгармас қийматини ўзгартириш мумкин:

```
enum Hafta {dush=8, sesh, chor=12, paysh=13, juma=16, shanba, yaksh=20};
```

Бу эълонда сеш қиймати 9, шанба эса 17 га тенг бўлади.

Санаб ўтилувчи ўзгармасларнинг номлари ҳар хил бўлиши керак, лекин уларнинг қийматлари бир хил бўлиши мумкин:

```
enum{nol=0, toza=0, bir, ikki, juft=2, uch};
```

Ўзгармаснинг қиймати ифода кўринишда берилиши мумкин, фақат ифодадаги номларнинг қийматлари шу қадамдагача аниқланган бўлиши керак:

```
enum {ikki=2,turt=ikki*2};
```

Ўзгармасни қийматлари манфий сон бўлиши ҳам мумкин:

```
enum {ikki=2,turt=ikki*2};
```

Мисол-1:

```
enum colors { BROWN, BLUE, RED, GREEN, YELLOW} ;
```

Юқорида фойдаланувчи томонидан янги солорс номни берилганларнинг тури аниқланди унинг қийматлари BROWN, BLUE, RED, GREEN ва YELLOW.

Мисол-2:

```
enum standing {FRESHMAN, SOPHOMORE, JUNIOR, SENIOR};
```

Юқорида фойдаланувчи томонидан янги standing номли санаб ўтилувчи тур аниқланди ва бу тур қуйидаги қийматларни қабул қилади. FRESHMAN, SOPHOMORE, JUNIOR ва SENIOR.

Қуйидаги мисолга эътибор беринг:

```
enum grades{'A', 'B', 'C', 'D', 'F'}; // берилганларнинг тақиқланган қийматлари
```

```
enum places {1ST, 2ND, 3RD, 4TH}; // берилганларнинг тақиқланган  
қийматлари
```

Юқоридаги мисолда ўзгарувчиларнинг янги аниқланган турлари бу қийматларни қабул қила олмайди сабаби улар идентификатор бўлиши керак.

```
enum grades {A, B, C, D, F};  
enum places{FIRST, SECOND, THIRD, FOURTH};
```

Агар берилганларни санаб ўтилувчи тури қабул қиладиган қийматлари ичидаги бирортаси олдин бошқа аниқланган тур қийматларида фойдаланилган бўлса унда ундан фойдаланиб бўлмайди.

Масалан:

```
enum mathStudent {JOHN, BILL, CINDY, LISA, RON};  
enum compStudent {SUSAN, CATHY, JOHN, WILLIAM}; //хатолик
```

Бу мисолда `compStudent` номни аниқланган турданинг қабул қиладиган қийматлари ичидаги `JOHN` қиймати олдин аниқланган `mathStudent` турида фойданалинган.

Дастурда фойдаланиладиган стандарт турдаги ўзгарувчилар қандай эълон қилинса `enum` туридаги ўзгарувчилар ҳам худди шундай эълон қилинади. Ўзгарувчиларни `enum` турида эълон қилиш синтаксиси қуйидагича:

```
dataType identifier, identifier,...;
```

Мисол: 

```
enum sports {BASKETBALL, FOOTBALL, HOCKEY, BASEBALL,  
SOCCER,VOLLEYBALL};
```

Юқорида `sports` номли санаб ўтилувчи тур аниқлаб олинди;

```
sports popularSport, mySport;
```

Юқорида `sports` турида бўлган `popularSport` ва `mySport` ўзгарувчилари эълон қилинди. Эълон қилинган ўзгарувчиларга қийматлар юклашимиз мумкин.

```
popularSport = FOOTBALL;  
mySport = popularSport;
```

`popularSport` номли ўзгарувчи `sports` тури қабул қилиши мумкин бўлган қийматлардан `FOOTBALL` қийматини ўзлаштирди. `mySport` номли ўзгарувчи `popularSport` ўзгарувчисининг қийматини ўзлаштириб олди.

## 16.2. Санаб ўтилувчи тур устида амаллар бажариш

Шуни айтиб ўтиш лозимки санаб ўтилувчи турлар устида қуйидаги арифметик амаллар бажариб бўлмайди. Қуйидаги мисолда кўрайлик:

```
mySport = popularSport + 2; //xatolik  
popularSport = FOOTBALL + SOCCER; //xatolik  
popularSport = popularSport * 2; //xatolik
```

Шунингдек инкремент ва декремент амаллари ҳам бажариб бўлмайди.

```
popularSport++; //xatolik  
popularSport--; //xatolik
```

Агар `popularSport` ўзгарувчисининг қийматини 1 га оширмакчи бўлсак, `static_cast` операторидан фойдаланишимиз мумкин. Қуйидаги мисолда келтирилган:

```
popularSport = static_cast<sports>(popularSport + 1);
```

Агар ўзгарувчига қиймат юкланган бўлса ва унинг қийматини 1 га оширсак, компилятор бу ўзгарувчига санаб ўтилувчи тур қабул қилиши мумкин бўлган қийматлар ичидан ўзи қабул қилган қийматдан кейинг қийматни юклайди.

Қуйида мисолда келтирилган:

```
popularSport = FOOTBALL;  
popularSport = static_cast<sports>(popularSport + 1);
```

Юқоридаги мисолда `popularSport` ўзгарувчисига `FOOTBALL` қиймати юкланди ва `popularSport` ўзгарувчисининг қиймати бирга оширилганда у `HOCKEY` қийматини ўзлаштирди.

```
popularSport = FOOTBALL;
popularSport = static_cast<sports>(popularSport - 1);
popularSport ўзгарувчиси BASKETBALL қийматини олди.
```

Таққослаш операторларидан фойдаланиш

Санаб ўтилувчи турдаги ўзгарувчилар қиймат қабул қилса демак бу ўзгарувчилар устида таққослаш операторларидан фойдаланиш мумкин. Қуйидаги мисолда спортс санаб ўтилувчи турида эълон қилинган `popularSport` ва `mySport` номли ўзгарувчилар устида таққослаш амаллари бажаришга доир мисол келтирилган:

```
FOOTBALL <= SOCCER // рост
HOCKEY > BASKETBALL // рост
BASEBALL < FOOTBALL // йолғон
popularSport = SOCCER;
mySport = VOLLEYBALL;
popularSport < mySport is true
```

Ўзгарувчиларни киритиш ва чиқариш

Берилганларни киритиш ва чиқариш фақат сатандарт инт, чар, доубле турдаги ўзгарувчилар учун ўринли, санаб ўтилувчи турдаги ўзгарувчиларга берилганларни тўғридан тўғри ўқиб, чиқариб бўлмайди.

```
enum courses {ALGEBRA, BASIC, PASCAL, CPP, PHILOSOPHY,
ANALYSIS, CHEMISTRY, HISTORY};
courses registered;
```

Юқоридаги келтирилган мисолда биринчи бўлиб `courses` номдаги санаб ўтилувчи тур аниқлаб олинди, иккинчи `courses` турида бўлган `registered` номли ўзгарувчи эълон қилинди. Санаб ўтилувчи турда эълон қилинган ўзгарувчини чар туридан фойдаланган ҳолда ўқиб олиш мумкин.

```
char ch1, ch2;
cin >> ch1 >> ch2; // чар туридаги иккита ўзгарувчини ўқиб олиш
```

```

switch (ch1)
{
case 'a':
case 'A':
    if (ch2 == 'l' || ch2 == 'L')
        registered = ALGEBRA;
    else
        registered = ANALYSIS;
    break;
case 'b':
case 'B':
    registered = BASIC;
    break;
case 'c':
case 'C':
    if (ch2 == 'h' || ch2 == 'H')
        registered = CHEMISTRY;
    else
        registered = CPP;
    break;
case 'h':
case 'H':
    registered = HISTORY;
    break;
case 'p':
case 'P':
    if (ch2 == 'a' || ch2 == 'A')
        registered = PASCAL;
    else
        registered = PHILOSOPHY;
    break;
default:
    cout << "Illegal input." << endl;
}

```



```
}
```

Худди шунингдек enum туридаги ўзгарувчини қуйидагича экранга чиқариш мумкин:

```
switch (registered)
{
case ALGEBRA:
    cout << "Algebra";
    break;
case ANALYSIS:
    cout << "Analysis";
    break;
case BASIC:
    cout << "Basic";
    break;
case CHEMISTRY:
    cout << "Chemistry";
    break;
case CPP:
    cout << "CPP";
    break;
case HISTORY:
    cout << "History";
    break;
case PASCAL:
    cout << "Pascal";
    break;
case PHILOSOPHY:
    cout << "Philosophy";
}
```

### 16.3. Функциялар ва enum турлар

Enum туридаги ўзгарувчиларни оддий турдаги ўзгарувчилар сингари функция параметри сифатида ишлатилиши мумкин. Худди шунингдек функция қайтарадиган қиймат тури сифатида ҳам фойдажанилиши мумкин. Қуйидаги

келтирилган мисолда берилганлар клавиатурадан киритилияпти ва enum туридаги қийматлар экранга чиқарилияпти.

```
courses readCourses()
{
courses registered;
char ch1, ch2;
cout << "Enter the first two letters of the course: "
    << endl;
cin >> ch1 >> ch2;
switch (ch1)
{
case 'a':
case 'A':
    if (ch2 == 'l' || ch2 == 'L')
        registered = ALGEBRA;
    else
        registered = ANALYSIS;
    break;
case 'b':
case 'B':
    registered = BASIC;
    break;
case 'c':
case 'C':
    if (ch2 == 'h' || ch2 == 'H')
        registered = CHEMISTRY;
    else
        registered = CPP;
    break;
case 'h':
case 'H':
    registered = HISTORY;
    break;
case 'p':
```

```

case 'P':
    if (ch2 == 'a' || ch2 == 'A')
        registered = PASCAL;
    else
        registered = PHILOSOPHY;
    break;
default:
    cout << "Illegal input." << endl;
}
return registered;
}
//end readCourse

```

Қуйидаги функция enum туридаги ўзгарувчи қийматини экранга чиқарияпти:

```

void printEnum(courses registered)
{
switch (registered)
{
case ALGEBRA:
cout << "Algebra";
break;
case ANALYSIS:
cout << "Analysis";
break;
case BASIC:
cout << "Basic";
break;
case CHEMISTRY:
cout << "Chemistry";
break;
case CPP:
cout << "CPP";
break;
case HISTORY:
cout << "History";

```

```

break;
case PASCAL:
cout << "Pascal";
break;
case PHILOSOPHY:
cout << "Philosophy";
}
//end switch
}
//end printEnum

```

#### 16.4. typedef билан ишлаш

Фойдаланувчи томонидан аниқланадиган тур **typedef** калит сўзи билан бошланади, ундан кейин мавжуд тур кўрсатилади ва идентификатор ёзилади. Охирида ёзилган идентификатор - янги яратилган турнинг номи ҳисобланади. Масалан,

```
typedef unsigned char byte;
```

ифодаси **byte** деб номланувчи янги турни яратади ва ўз мазмунига кўра **unsigned char** тури билан эквивалент бўлади. Кейинчалик, программада хотирадан бир байт жой эгаллайдиган ва [0..255] оралиқдаги қийматларни қабул қиладиган **byte** туридаги ўзгарувчи (ўзгармасларни) эълон қилиш мумкин:

```
byte c=65;
byte Byte=0xFF;
```

Массив кўринишидаги фойдаланувчи томонидан аниқланувчи тур эълони қуйидагича бўлади:

```
typedef char Ism[30];
Ism ism;
```

Исм туридаги исм ўзгарувчиси эълони - бу 30 белгидан иборат массив (сатр) эълонидир.

Одатда эчилаётган масаланинг предмет соҳаси терминларида ишлаш учун структуралар қайта номланади. Натижада мураккаб тузилишга эга бўлган

ва зарур хусусиятларни ўзига жамлаган янги турларни яратишга мувофик бўлинади.

Масалан, комплекс сон ҳақидаги маълумотларни ўз ичига оловчи Complex тури қуйидагича аниқланади:

```
typedef struct
{
    double re;
    double im;
} Complex;
```

Енди комплекс сон эълонини

```
Complex KSon;
```

ёзиш мумкин ва унинг майдонларига мурожаат қилиш мумкин:

```
KSon.re=5.64;
```

```
KSon.im=2.3;
```

#### **Адабиётлар рўйхати:**

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.- М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

#### **Мустақил таълим мавзулари:**

1. Массивларни typedef орқали тасвирлаш.
2. Фойдаланувчи томонидан аниқланган берилганлар турларини функцияларда ишлатиш.
3. Санаб ўтилувчи турнинг оддий ўзгармаслардан фарқи.

## Глоссарий

Термин	Terminology	Ўзбек тилидаги шархи
include	include	препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади.
typedef	typedef	турларни янги ном билан ишлатиш имконини беради
enum	enum	санаб ўтилувчи тур
аргумент	Argument	функцияга параметрига жўнатиладиган қиймат
ўзгармас	Variable	дастур ишлаши давомида қияматини ўзгартрмайдиган берилган
сарлавҳа файли	Header file	функциялар эълони ёзилган файл
switch	switch	танлаш оператори
cout	cout	экранга чиқариш оқими
cin	cin	экрандан киритиш оқими
if	if	шарт оператори
else	else	шарт ёлғонлигини аниқловчи оператор

## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур векторни ўсиш, камайиш бўйича тартиблаб чоп қилувчи, шунингдек максимум элемент жойлашган индексни топиш имконини берувчи функциялардан ташкил топган. Массивни функцияга жўнатишда typedef ёрдамида массив учун яратилган янги турдан фойдаланилган. Дастур ишлаши натижасида максимум элемент жойлашган индексни топишда нотўғри натижа чиқмоқда. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили имкониятларидан келиб чиқиб тўғри натижа берувчи дастур кодини тақдим этинг (индивидуал ҳолда).

## Назорат саволлари

1. Санаб ўтилувчи турлар нима мақсадда ишлатилади?
2. Санаб ўтилувчи турни аниқлаш қандай қисмлардан иборат?
3. Санаб ўтилувчи тур қандай хоссаларга эга?
4. Санаб ўтилувчи турлар устида қандай амаллар бажариб бўлмайди?
5. Санаб ўтилувчи турлар устида амал бажаришга мисол келтиринг.
6. Санаб ўтилувчи турга берилганларни киритиш ва чиқариш қандай ўзгарувчилар учун ўринли?
7. Enum туридаги ўзгарувчилардан қандай мақсадларда фойдаланиш мумкин?
8. enum туридаги ўзгарувчи эълонига мисол келтиринг.
9. Санаб ўтилувчи турлар устида таққослаш амалига мисол келтиринг.
10. typedef калит сўзи ёрдамида янги тур хосил қилишга мисол келтиринг.

## Тест саволлари

1. Санаб ўтилувчи тур тўғри кўрсатилган жавобни кўрсатинг.
  - a) enum
  - b) typedef
  - c) struct
  - d) class
2. Турни бошқа ном билан аташ учун қайси калит сўз ишлатилади?
  - a) typedef
  - b) enum
  - c) function
  - d) static
3. shanba номли ўзгармаснинг қиймати неччига тенг?  
enum Hafta {dush=8, sesh, chor=12, paysh=13, juma=16, shanba, yaksh=20};
  - a) 17
  - b) 15
  - c) 19
  - d) 20

4. `paysh` номли ўзгармаснинг қиймати неччига тенг?

```
enum Hafta {dush=8, sesh, chor=12, paysh=1, juma=16, shanba, yaksh=20};
```

- a) 1
- b) 13
- c) 15
- d) 20

5. Қуйидаги дастур қисми бажарилиши натижасида экранга нима чиқади?

```
enum Hafta {dush=8, sesh, chor=12, paysh=1, juma=16, shanba, yaksh=20};
```

```
enum Oy {yanvar=8, fevral, mart=12, aprel=1, may=16};
```

```
Hafta a;
```

```
Oy b;
```

```
a=sesh;
```

```
b=fevral;
```

```
cout << a+b;
```

- a) 18
- b) 0
- c) 2
- d) 22

6. Қуйидаги дастур қисми бажарилиши натижасида экранга нима чиқади?

```
enum Hafta {dush=8, sesh, chor, paysh, juma, shanba, yaksh};
```

```
enum Oy {yanvar=8, fevral, mart, aprel, may};
```

```
Hafta a;
```

```
Oy b;
```

```
a=juma;
```

```
b=may;
```

```
cout << a+b;
```

- a) 24
- b) 6
- c) 7
- d) 12



7. Қуйидаги дастур қисмидаги хато кодни топинг.

```
enum Hafta {dush=8, sesh, chor, paysh, juma, shanba, yaksh};  
Hafta a;  
a=juma;  
a++;
```

- a) a++;
- b) dush=8
- c) Hafta a;
- d) a=juma;

8. Дастур қисми охирида popularSport ўзгарувчиси қайси константани олади.

```
enum sports {BASKETBALL, FOOTBALL, HOCKEY, BASEBALL,  
SOCCER, VOLLEYBALL};  
sports popularSport = FOOTBALL;  
popularSport = static_cast<sports>(popularSport + 1);
```

- a) HOCKEY
- b) FOOTBALL
- c) BASEBALL
- d) BASKETBALL

9. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
bool c;  
int a=5, b=2;  
c=(a!=b) + (a>b);  
cout << c ;
```

- a) 1
- b) 2
- c) 0
- d) 3

10. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
short a=5;
```

```
cout << sizeof(a) << ' ' << sizeof(d);
```

- a) 2 4
- b) 4 2
- c) 5 123
- d) 2

## Маъруза 17. Стандарт кутубхона функциялари

### Режа:

1. cctype кутубхона файли
2. Математик функциялар кутубхонаси
3. Белгилар билан ишлаш функциялари
4. Турларни ўзгартириш функциялари

**Таянч иборалар:** *include, кутубхона, isalpha, isdigit, isalnum, cout, cin, Ўзгарувчи, Идентификатор*

C++ нинг стандарт кутубхонаси кўпгина аниқланган функциялар, ўзгармаслар ва берилганларнинг махсус турларидан ташкил топган.

### 17.1. cctype (ctype.h) кутубхона файли

Қуйидаги жадвалда cctype(ctype.h) кутубхона файлининг стандарт функциялари келтирилган:

Функция номлари ва параметрлари	Параметрлар турлари	Функция қайтарадиган қиймат
isalnum(ch)	ch bu char turidagi o'zgaruvchi	Функция қуйидаги бутун турдаги қийматларни қайтаради: <ul style="list-style-type: none"><li>• агар ch ўзгарувчи қиймати 'A'-'Z', 'a'-'z', '0'-'9' оралиғида бўлса 1 (true);</li><li>• акс ҳолда 0 (false) қиймат қайтаради;</li></ul>
isctrl(ch)	ch bu char turidagi o'zgaruvchi	Функция қуйидаги бутун турдаги қийматларни

		<p>қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи қиймати ASCII жадвалидаги 0-31 ва 127 ) бўлса 1(true);</li> <li>• акс ҳолда 0 (false);</li> </ul>
isdigit(ch)	ch bu char turidagi o'zgaruvchi	<p>Функция қуйидаги бутун турдаги қийматларни қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи қиймати '0' - '9' белгилар белгиларини қабул қилса 1(true);</li> <li>• акс ҳолда, 0(false);</li> </ul>
islower(ch)	ch bu char turidagi o'zgaruvchi	<p>Функция қуйидаги бутун турдаги қийматларни қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи қиймати 'a' - 'z' оралиғидаги белгилардан бири бўлса, нол бўлмаган қиймат (true) қайтаради</li> <li>• акс ҳолда, 0(false);</li> </ul>
isprint(ch)	ch bu char turidagi o'zgaruvchi	<p>Функция қуйидаги бутун турдаги қийматларни қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи қиймати ASCII жадвалидаги, пробел ёки “~” қийматни қабул қилса,</li> </ul>

		<p>1 (true);</p> <ul style="list-style-type: none"> <li>• акс холда, 0(false);</li> </ul>
ispunct(ch)	ch bu char turidagi o'zgaruvchi	<p>Функция қуйидаги бутун турдаги қийматларни қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи қиймати пунктуация белгиларини қабул қилса 1 (true);</li> <li>• акс холда, 0(false);</li> </ul>
isspace(ch)	ch bu char turidagi o'zgaruvchi	<p>Функция қуйидаги бутун турдаги қийматларни қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи қиймати (пробел, янги қатор, таб) белгиларидан бири бўлса нол бўлмаган қиймат (true) қайтаради;</li> <li>• акс холда, 0(false);</li> </ul>
isupper(ch)	ch bu char turidagi o'zgaruvchi	<p>Функция қуйидаги бутун турдаги қийматларни қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи қиймати катта ҳарфларни ('A'-'Z' ) қабул қилса 1 (true);</li> <li>• акс холда, 0(false);</li> </ul>

## 17.2. Математик функциялар кутубхонаси (math.h)

Функция прототипи	Бажарадиган амали
int abs(int i)	i сонни абсолют қийматини қайтаради
double acos(double x)	Радианда берилган x аргументни арккосинус қийматини қайтаради
double asin(double x)	Радианда берилган x аргументни арксинус қийматини қайтаради
double atan(double x)	Радианда берилган x аргументни арктангенс қийматини қайтаради
double atan2(double x, double y)	Радианда берилган x/y нисбатнинг арктангенси қийматини қайтаради
double ceil(double x)	Ҳақиқий x қийматини унга энг яқин катта бутун сонгача айлантиради ва уни хақиқий кўринишда қайтаради
double cos(double x)	x радианга тенг бўлган бурчакни косинусини қайтаради
double cosh(double x)	x радианга тенг бўлган бурчакни гиперболик косинусини қайтаради
double exp(double x)	$e^x$ қийматни қайтаради
double fabs(double x)	Ҳақиқий сонни абсолют қийматини Қайтаради
double floor(double x)	Ҳақиқий x қийматни энг яқин кичик сонга айлантиради ва уни ҳақиқий сон кўринишида қайтаради

double fmod(double x, double y )	х сонини у сонига бўлиш натижасидаги қолдиқни қайтаради. % амалига ўхшаган, фақат ҳақиқий сон қайтаради
double frexpr(double x, int *expPtr)	х сонни мантиссасини ва даражасини ажратиб, мантисса қийматини қайтаради ва даражасини кўрсатилган expPtr адресига жойлаштиради
double hypot(double x, double y)	Тўғри учбурчакни катетлари бўйича гипотенузани ҳисоблайди
long int labs(long int num)	num узун бутун соннинг абсолют қийматини қайтаради
double ldexp(double x, int exp)	$X \cdot 2^{\text{exp}}$ қийматни қайтаради
double log(double x)	х сонининг натурал логарифминини қайтаради

math standart kutubxonasidagi funksiyalardan foydalanish uchun cmath sarlovha faylini qo'shib qo'yish kerak.

Misol:

```
#include <iostream>
#include <cmath>
#include <cctype>
using namespace std;
int main()
```

```

{
int x;
double u, v;
cout << "Line 1: Uppercase a is "
      << static_cast<char>(toupper('a'))
      << endl; //Line 1
u = 4.2;      //Line 2
v = 3.0;     //Line 3
cout << "Line 4: " << u << " to the power of "
      << v << " = " << pow(u, v) << endl; //Line 4
cout << "Line 5: 5.0 to the power of 4 = "
      << pow(5.0, 4) << endl; //Line 5
u = u + pow(3.0, 3); //Line 6
cout << "Line 7: u = " << u << endl; //Line 7
x = -15; //Line 8
cout << "Line 9: Absolute value of " << x
      << " = " << abs(x) << endl; //Line 9
return 0;
}

```

Дастур ишлашининг натижаси:

Line 1: Uppercase a is A

Line 4: 4.2 to the power of 3 = 74.088

Line 5: 5.0 to the power of 4 = 625

Line 7: u = 31.2

Line 9: Absolute value of -15 = 15



### 17.3. Белгилар билан ишлаш функциялари

Программалаш амалиётида белгиларни қайсидир ораликқа тегишли эканлигини билиш зарур бўлади. Буни «ctype.h» сарлавҳа файлида эълон қилинган функциялар ёрдамида аниқлаш мумкин.

Қуйида уларнинг бир қисмининг тавсифи келтирилган:

isalnum() - белги рақам ёки ҳарф (true) ёки йўқлигини (false) аниқлайди;

isalpha() - белгини ҳарф (true) ёки йўқлигини (false) аниқлайди;

isascii() - белгини коди 0..127 оралиғида (true) ёки йўқлигини (false) аниқлайди;

isdigit() - белгини рақамлар диапазонида тегишли (true) ёки йўқлигини (false) аниқлайди.

Бу функциялардан фойдаланишга мисол келтирамиз.

```
#include <iostream>
#include <ctype.h>
#include <string.h>
using namespace std;
int main()
{
    char satr[5];
    int xato;
    do
    {
        xato=0;
        cout<<"\nTug'ilgan yilingizni kiriting: ";
        cin.getline(satr,5);
        for (int i=0; i<strlen(satr) && !xato; i++)
        {
```

```

if(isalpha(satr[i]))
{
    cout<<"Harf kiritdildi!";
    xato=1;
}
else
    if(iscntrl(satr[i]))
    {
        cout<<"Boshqaruv belgisi kiritildi!";
        xato=1;
    }
    else
        if(ispunct(satr[i]))
        {
            cout<<"Punktuatsiya belgisi kiritildi!";
            xato=1;
        }
        else
            if (!isdigit(satr[i]))
            {
                cout<<"Raqamdan farqli belgi kiritdildi!";
                xato=1;
            }
        }
    if (!xato)
    {
        cout << "Sizni tug'ilgan yilingiz: "<<satr;
    }

```

```

    return 0;
}
} while (1);
}

```

Программада фойдаланувчига туғилган йилини киритиш таклиф этилади. Киритилган сана satr ўзгарувчисига ўқилади ва агар сатрнинг ҳар бир белгиси (satr[i]) ҳарф ёки бошқарув белгиси ёки пунктуация белгиси бўлса, шу ҳақда хабар берилади ва туғилган йилни қайта киритиш таклиф этилади. Программа туғилган йил ( тўртта рақам) тўғри киритилганда “Sizni tug'ilgan yilingiz: XXXX” сатрини чоп қилиш билан ўз ишини тугатади.

#### 17.4. Турларни ўзгартириш функциялари

Сатрлар билан ишлашда сатр кўринишида берилган сонларни, сон турларидаги қийматларга айлантириш ёки тескари амални бажаришга тўғри келади. C++ тилининг «stdlib.h» кутубхонасида бу амалларни бажарувчи функциялар тўплами мавжуд. Қуйида нисбатан кўп ишлатиладиган функциялар тавсифи келтирилган.

atoi( ) функциясининг синтаксиси

```
int atoi(const char* ptr);
```

кўринишга эга бўлиб, ptr кўрсатувчи ASCIIZ-сатрни int туридаги сонга ўтказишни амалга оширади. Функция сатр бошидан белгиларни сонга айлантира бошлайди ва сатр охиригача ёки биринчи рақам бўлмаган белгигача ишлайди. Агар сатр бошида сонга айлантириш мумкин бўлмаган белги бўлса, функция 0 қийматини қайтаради. Лекин, шунга эътибор бериш керакки, “0” сатри учун ҳам функция 0 қайтаради. Агар сатрни сонга айлантиришдаги ҳосил бўлган сон int чегарасидан чиқиб кетса, соннинг кичик икки байти натижа сифатида қайтарилади.

Мисол учун

```
#include <stdlib.h>
```

```
#include <iostream.h>
```

```

int main()
{
    char str[]="32secund";
    int i=atoi(str);
    cout<<i<<endl;
    return 0;
}

```

программасининг натижаси сифатида экранга 32 сонини чоп этади. Агар str қиймати "100000" бўлса, экранга -31072 қиймати чоп этилади, чунки 100000 сонинг ички кўриниши 0x186A0 ва унинг охириги икки байтидаги 0x86A0 қиймати 31072 сонининг қўшимча коддаги кўринишидир.

atol() функцияси худди atoi() функциясидек амал қилади, фақат функция натижаси long турида бўлади. Агар ҳосил бўлган сон қиймати long чегарасига сиғмаса, функция кутилмаган қийматни қайтаради.

atof() функцияси эълони

```
double atof (const char* ptr);
```

кўринишида бўлиб, ptr кўрсатувчи ASCIIZ-сатрни double туридаги сузувчи нуқтали сонга ўтказишни амалга оширади. Сатр сузувчи нуқтали сон форматида бўлиши керак. Сонга айлантириш биринчи форматга мос келмайдиган белги учрагунча ёки сатр охиригача давом этади.

strtod() функцияси atof() функциясидан фарқли равишда сатрни double туридаги сонга ўтказишда конвертация жараёни узилган пайтда айлантириш мумкин бўлмаган биринчи белги адресини ҳам қайтаради. Бу ўз навбатида сатрни хато қисмини қайта ишлаш имконини беради.

strtod() функциясининг синтаксиси

```
double strtod(const char *s, char **endptr);
```

кўринишга эга ва endptr кўрсаткичи конвертация қилиниши мумкин бўлмаган биринчи белги адреси.

itoa() va ltoa() функциялари мос равишда int va long туридаги сонларни сатрга кўринишга ўтказди. Бу функциялар мос равишда қуйидаги синтаксисга эга:

```
char* itoa(int num, char *str, int radix);
```

ва

```
char* ltoa(long num, char *str, int radix);
```

Бу функциялар num сонини radix аргументда кўрсатилган санок системасидаги кўринишини str сатрда ҳосил қилади.

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)

2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)

3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)

4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим мавзулари:

1. conio.h кутубхонаси функциялари билан ишлаш.
2. Белгилар билан ишлаш функциялари

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include	include	препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
кутубхона	library	дастурга <b>include</b> директиваси ёрдамида қўшиладиган файллар

isalpha	isalpha	белгини харф ёки йўқлигини аниқлайдиган функция
isdigit	isdigit	белгини рақамлар диапазонида тегишлилигини аниқлайдиган функция
isalnum	isalnum	белги рақам ёки ҳарфлигини аниқлайдиган функция
cout	cout	экранга чиқариш оқими
cin	cin	экрандан киритиш оқими
Ўзгарувчи	Variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
Идентификатор	identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизик ('_') белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги

### Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур бутун сонлар бўлинмасининг қолдиқ қисмини топади. Соннинг тури ҳақиқий сон турига ўтказилса дастур компиляция хатолигини юзага келтирмоқда. Энди ушбу масалани ҳақиқий сон учун ишлайдиган қилиш керак.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

### Назорат саволлари:

1. `cctype` (`ctype.h`) кутубхона файлининг қандай функцияларини биласиз?
2. `isdigit()` функцияси қандай вазифани бажаради?
3. `isspace()` функцияси қандай вазифани бажаради?
4. `cctype` (`ctype.h`) кутубхонаси функцияларининг ишлатилишига мисоллар келтиринг.
5. Математик функциялардан фойдаланиш учун қандай кутубхона қўшиш керак?
6. `math.h` кутубхонасидаги қандай функцияларини биласиз?
7. `abs()` функцияси қандай вазифани бажаради?
8. Турларни ўзгартирувчи функциялар қайси кутубхонада жойлашган?
9. Турларни ўзгартирувчи қандай функциялар мавжуд?
10. `atoi` функциясини ишлатишга мисол келтиринг.

### Тест саволлари

1.  $|x|$  C++ да ҳақиқий сонлар учун қандай ёзилади?
  - a) `fabs(x)`
  - b) `abs(x)`
  - c) `labs(x)`
  - d) `absf(x)`
2. 10 даражаси  $x$  C++ да қандай ёзилади?
  - a) `pow10(x)`
  - b) `pow(10,x)`
  - c) `pow(x,10)`
  - d)  $10^x$
3. Оқимдан сатрни ўқишга мўлжалланган `gets()` функциясининг прототипи тўғри ёзилган жавобни танланг
  - a) `char* gets(char *s);`
  - b) `char gets(char *s);`
  - c) `char* gets(char s);`
  - d) `string* gets(string *s);`

4. C++ тилида char туридаги сатрни нусхаловчи функцияни аниқланг.

- a) strcpy()
- b) strcat()
- c) strcmp()
- d) strrev()

5. C++ тилида ихтиёрий турнинг хотирадаги ўлчамини аниқлаш функциясини кўрсатинг

- a) sizeof
- b) length
- c) strlen
- d) size

6. C++ тилида x сонинг абсолют қийматини аниқлаш функциясини кўрсатинг

- a) abs(x)
- b) fabs(x)
- c) exp(x)
- d) ceil(x)

7. C++ тилида сатр узунлигини қайтарувчи функцияни аниқлинг.

- a) strlen()
- b) malloc()
- c) ceil()
- d) To'g'ri javob yo'q

8.  $\sin 2x + 2\cos y$  ифодаси c++ да қандай ифодаланади?

- a)  $\sin(2*x)+2*\cos(y)$
- b)  $\sin(2x)+2\cos(y)$
- c)  $\sin 2*x +2*\cos y$
- d)  $\sin*2*x+2*\cos*y$



9.  $\arcsin 2x + 2\arccos y$  ифодаси с++ да қандай ифодаланади?

a)  $\arcsin(2x) + 2\arccos(y)$

b)  $\sin(2x) + 2\arccos(y)$

c)  $\arcsin(2x) + 2\cos(y)$

d)  $\sin^2 x + 2\cos y$

10.  $\ln 2x + \ln 2\cos y$  ифодаси с++ да қандай ифодаланади?

a)  $\log(2x) + \log(2\cos(y))$

b)  $\ln(2x) + \ln(2\arccos(y))$

c)  $\lg(2x) + \lg(2\arccos(y))$

d)  $\log(2x) + \ln(2\arccos(y))$

## Маъруза 18. Кўрсаткичлар ва адрес олувчи ўзгарувчилар

### Режа:

1. Кўрсаткичлар
2. Кўрсаткичга бошланғич қиймат бериш
3. Кўрсаткич устида амаллар
4. Адресни олиш амали
5. Кўрсаткичлар ва адрес олувчи ўзгарувчилар функция параметри сифатида

### 18.1. Кўрсаткичлар

Дастур матнида ўзгарувчи эълон қилинганда, компилятор ўзгарувчига хотирадан жой ажратади. Бошқача айтганда, дастур коди хотирага юкланганда берилганлар учун, улар жойлашадиган сегментнинг бошига нисбатан силжишини, яъни нисбий адресини аниқлайди ва объект код ҳосил қилишда ўзгарувчи учраган жойга унинг адресини жойлаштиради.

Умуман олганда, дастурдаги ўзгармаслар, ўзгарувчилар, функциялар ва синф объектлар адресларини хотиранинг алоҳида жойида сақлаш ва улар устидан амаллар бажариш мумкин. Қийматлари адрес бўлган ўзгарувчиларга *кўрсаткич ўзгарувчилар* дейилади.

Кўрсаткич уч хил турда бўлиши мумкин:

- бирорта объектга, хусусан ўзгарувчига кўрсаткич;
- функцияга кўрсаткич;
- воид кўрсаткич.

Кўрсаткичнинг бу хусусиятлари унинг қабул қилиши мумкин бўлган қийматларида фарқланади.

Кўрсаткич албатта бирорта турга боғланган бўлиши керак, яъни у кўрсатган адресда қандайдир қиймат жойланиши мумкин ва бу қийматнинг хотирада қанча жой эгаллаши олдиндан маълум бўлиши шарт.

**Функцияга кўрсаткич.** Функцияга кўрсаткич дастур жойлашган хотирадаги функция кодининг бошланғич адресини кўрсатади, яъни функция чақирилганда бошқарув айна шу адресга узатилади. Кўрсаткич орқали функцияни оддий ёки воситали чақириш амалга ошириш мумкин. Бунда

функция унинг номи бўйича эмас, балки функцияга кўрсатувчи ўзгарувчи орқали чақирилади. Функцияни бошқа функцияга аргумент сифатида узатиш ҳам функция кўрсаткичи орқали бажарилади. Функцияга кўрсаткичнинг ёзилиш синтаксиси қуйидагича:

`<тур> (* <ном>) (<параметрлар рўйхати>);`

Бунда `<тур>`- функция қайтарувчи қиймат тури; `*<ном>` - кўрсаткич ўзгарувчининг номи; `<параметрлар рўйхати>` - функция параметрларининг ёки уларнинг турларининг рўйхати.

Масалан:

```
int (*fun) (float, float) ;
```

Бу ерда бутун сон турида қиймат қайтарадиган фун номидаги функцияга кўрсаткич эълон қилинган ва у иккита ҳақиқий турдаги параметрларга эга.

**Масала.** Берилган бутун  $n=100$  ва  $a, b$  - ҳақиқий сонлар учун  $f_1(x) = 5 \sin(3x) + x$ ,  $f_2(x) = \cos(x)$  ва  $f_3(x) = x^2 + 1$  функциялар учун  $\int_a^b f(x) dx$  интегрални тўғри тўртбурчаклар формуласи билан тақрибан ҳисоблансин:

$$\int_a^b f(x) dx \approx h[f(x_1) + f(x_2) + \dots + f(x_n)],$$

бу ерда  $h = \frac{b-a}{n}$ ,  $x_i = a + ih - h/2, i = 1..n$ .

Дастур бош функция, интеграл ҳисоблаш ва иккита математик функциялар -  $f_1(x)$  ва  $f_3(x)$  учун аниқланган функциялардан ташкил топади,  $f_2(x) = \cos(x)$  функциянинг адреси «`cmath.h`» сарлавха файлидан олинади. Интеграл ҳисоблаш функциясига кўрсаткич орқали интеграл ҳисобланадиган функция адреси,  $a$  ва  $b$  - интеграл чегаралари қийматлари узатилади. Ораликни бўлишлар сони -  $n$  глобал ўзгармас қилиб эълон қилинади.

```
#include <iostream.h>
#include <math.h>
const int n=100;
double f1(double x){return 5*sin(3*x)+x;}
double f3(double x){return x*x+1;}
```

```

double Integral(double(*f)(double),double a,double b)
{
    double x,s=0;
    double h=(b-a)/n;
    x=a-h/2;
    for(int i=1;i<=n; i++) s+=f(x+=h);
    s*=h;
    return s;
}
int main()
{
    double a,b;
    int menu;
    while(1)
    {
        cout<<"\nIsh rejimini tanlang:\n";
        cout<<"1:f1(x)=5*sin(3*x)+x integralini\
hisoblash\n";
        cout<<"2:f2(x)=cos(x) integralini hisoblash\n";
        cout<<"3:f3(x)=x^2+1 integralini hisoblash\n";
        cout<<"0:Dasturdan chiqish\n";
        do
        {
            cout<<" Ish rejimi-> ";
            cin>>menu;
        }
        while (menu<0 || menu>3);
        if(!menu)break;
        cout<<"Integral oralig'ining quyi chegarasi a=";
        cin>>a;
        cout<<"Integral oralig'ining yuqori chegarasi b=";
        cin>>b;
        cout<<"Funksiya integrali S=";
        switch (menu)

```

```

    {
        case 1 : cout<<Integral(f1,a,b)<<endl; break;
        case 2 : cout<<Integral(cos,a,b)<<endl; break;
        case 3 : cout<<Integral(f3,a,b)<<endl;
    }
}
return 0;
}

```

Дастурнинг иши чексиз такрорлаш оператори танасини бажаришдан иборат. Такрорлаш танасида фойдаланувчига иш режимини танлаш бўйича меню таклиф қилинади:

Ish rejimini tanlang:

1:  $f_1(x)=5*\sin(3*x)+x$  integralini hisoblash

2:  $f_2(x)=\cos(x)$  integralini hisoblash

3:  $f_3(x)=x^2+1$  integralini hisoblash

0: Dasturdan chiqish

Ish rejimi->

Фойдаланувчи 0 ва 3 оралиғидаги бутун сонни киритиши керак. Агар киритилган сон (**menu** ўзгарувчи қиймати) 0 бўлса, break оператори ёрдамида такрорлашдан, кейин дастурдан чиқилади. Агар **menu** қиймати 1 ва 3 оралиғида бўлса, интегралнинг қуйи ва юқори чегараларини киритиш сўралади, ҳамда Интеграл() функцияси мос функция адреси билан чақирилади ва натижа чоп этилади. Шунга эътибор бериш керакки, интеграл чегараларининг қийматларини тўғри киритилишига фойдаланувчи жавобгар.

**Объектга кўрсаткич.** Бирор объектга кўрсаткич (шу жумладан ўзгарувчига). Бундай кўрсаткичда маълум турдаги (таянч ёки ҳосилавий турдаги) берилганларнинг хотирадаги адреси жойлашади. Объектга кўрсаткич куйидагича эълон қилинади:

<тур> \*<ном>;

Бу ерда <тур> - кўрсаткич аниқлайдиган адресдаги қийматнинг тури, <ном> - объект номи (идентификатор). Агар бир турда бир нечта кўрсаткичлар эълон қилинадиган бўлса, ҳар бир кўрсаткич учун ‘\*’ белгиси қўйилиши шарт:

```
int *i, j,*k;
float x,*y,*z;
```

Келтирилган мисолда **i** ва **k** - бутун турдаги кўрсаткичлар ва **j** - бутун турдаги ўзгарувчи, иккинчи операторда **x** - ҳақиқий ўзгарувчи ва **y**, **z** - ҳақиқий турдаги кўрсаткичлар эълон қилинган.

**void кўрсаткич.** Бу кўрсаткич объект тури олдиндан номаълум бўлганда ишлатилади. void кўрсаткичининг муҳим афзалликларидан бири - унга ҳар қандай турдаги кўрсаткич қийматини юклаш мумкинлигидир. void кўрсаткич адресидаги қийматни ишлатишдан олдин, уни аниқ бир турга ошкор равишда келтириш керак бўлади. void кўрсаткични эълон қилиш куйидагича бўлади:

```
void *<ном>;
```

Кўрсаткичнинг ўзи ўзгармас ёки ўзгарувчан бўлиши ва ўзгармас ёки ўзгарувчилар адресига кўрсатиши мумкин, масалан:

```
int i;           // butun o'zgaruvchi
const int ci=1; // butun o'zgarmas
int * pi;        // butun o'zgaruvchiga ko'rsatkich
const int *pci; // butun o'zgarmasga ko'rsatkich
int *const cp=&i; // butun o'zgaruvchiga o'zgarmas
                // ko'rsatkich
const int*const cpc=&ci; // butun o'zgarmasga
                        // o'zgarmas ko'rsatkich
```

Мисоллардан кўриниб турибдики, '\*' ва кўрсаткич номи орасида турган const модификатори фақат кўрсаткичнинг ўзига тегишли ҳисобланади ва уни ўзгартириш мумкин эмаслигини билдиради, '\*' белгисидан чапда турган const эса кўрсатилган адресдаги қиймат ўзгармас эканлигини билдиради.

Кўрсаткичга қийматни бериш учун '&' - адресни олиш амали ишлатилади.

Кўрсаткич ўзгарувчиларининг амал қилиш соҳаси, яшаш даври ва кўриниш соҳаси умумий қоидаларга бўйсунди.

## 18.2. Кўрсаткичга бошланғич қиймат бериш

Кўрсаткичлар кўпинча динамик хотира (бошқача номи «уюм» ёки «хеап») билан боғлиқ ҳолда ишлатилади. Хотиранинг динамик дейилишига

сабаб, бу соҳадаги бўш хотира дастур ишлаш жараёнида, керакли пайтида ажратиб олинади ва зарурат қолмаганида қайтарилади (бўшатилади). Кейинчалик, бу хотира бўлаги дастур томонидан бошқа мақсадда яна ишлатилиши мумкин. Динамик хотирага фақат кўрсаткичлар ёрдамида мурожаат қилиш мумкин. Бундай ўзгарувчилар *динамик ўзгарувчилар* дейилади ва уларни яшаш вақти яратилган нуқтадан бошлаб дастур охиригача ёки ошкор равишда йўқотилган (боғланган хотира бўшатиладиган) жойгача бўлади.

Кўрсаткичларни эълон қилишда унга бошланғич қийматлар бериш мумкин. Бошланғич қиймат (инициализатор) кўрсаткич номидан сўнг ёки қавс ичида ёки '=' белгидан кейин берилади. Бошланғич қийматлар куйидаги усуллар билан берилиши мумкин:

Кўрсаткичга мавжуд бўлган объектнинг адресини бериш:

a) адресни олиш амал орқали:

```
int i=5,k=4; // butun o'zgaruvchilar
int *p=&i;    // p ko'rsatkichga i o'zgaruvchining
              // adresi yoziladi
int *p1(&k); // p1 ko'rsatkichga k o'zgaruvchining
              // adresi yoziladi
```

b) бошқа, инициализацияланган кўрсаткич қийматини бериш:

```
int * r=p; // p oldin e'lon qilingan va qiymatga ega
           // bo'lgan ko'rsatkich
```

d) массив ёки функция номини бериш:

```
int b[10]; // massivni e'lon qilish
int *t=b; // massivning boshlang'ich adresini berish
void f(int a){/* ... */} // funksiyani aniqlash
void (*pf)(int); // funksiyaga ko'rsatkichni e'lon qilish
pf=f; // funksiya adresini ko'rsatkichga berish
```

Ошкор равишда хотиранинг абсолют адресини бериш:

```
char *vp = (char *)0xB8000000;
```

Бунда `0xB8000000` - ўн олтилик ўзгармас сон ва (`char *`) - турга келтириш амали бўлиб, у вп ўзгарувчисини хотиранинг абсолют адресидаги байтларни чар сифатида қайта ишловчи кўрсаткич турига айлантеририлишини англатади.

Бўш қиймат бериш:

```
int *suxx=NULL;
int *r=0;
```

Биринчи сатрда махсус НУЛЛ ўзгармаси ишлатилган, иккинчи сатрда 0 қиймат ишлатилган. Иккала ҳолда ҳам кўрсаткич ҳеч қандай объектга мурожаат қилмайди. Бўш кўрсаткич асосан кўрсаткични аниқ бир объектга кўрсатаётган ёки йўқлигини аниқлаш учун ишлатилади.

### 18.3. Кўрсаткич устида амаллар

Кўрсаткич устида қуйидаги амаллар бажарилиши мумкин:

- 1) объектга воситали мурожаат қилиш амали;
- 2) қиймат бериш амали;
- 3) кўрсаткичга ўзгармас қийматни кўшиш амали;
- 4) айириш амали;
- 5) инкремент ва декремент амаллари;
- 6) солиштириш амали;
- 7) турга келтириш амали.

Воситали мурожаат қилиш амали кўрсаткичдаги адрес бўйича жойлашган қийматни олиш ёки қиймат бериш учун ишлатилади:

```
char a; // char turidagi o'zgaruvchi e'loni.
char *p=new char; // Ko'rsatkichni e'lon qilib, unga
// dinamik xotiradan ajratilgan
// xotiraning adresini berish
*p='b'; // p adresiga qiymat joylashtirish
a=*p; // a o'zgaruvchisiga p adresni berish
```

Шуни қайд қилиб ўтиш керакки, хотиранинг аниқ бир жойидаги адресни бир пайтнинг ўзида бир нечта ва ҳар хил турдаги кўрсаткичларга бериш мумкин ва улар орқали мурожаат қилинганда берилганнинг ҳар хил турдаги қийматларини олиш мумкин:

```
unsigned long int A=0Xcc77ffaa;
```



```

unsigned short int * pint=(unsigned short int*)&A;
unsigned char* pchar=(unsigned char*)&A;
cout<<hex<<A<<' '<<hex<<*pint<<' '<<hex<<(int)*pchar;

```

Экранга ҳар хил қийматлар чоп этилади:

```
cc77ffaa ffaa aa
```

Ўзгарувчилар битта адресда жойлашган ҳолда яхлит қийматнинг турли бўлақларини ўзлаштиради. Бунда, бир байтдан катта жой эгаллаган сон қийматининг хотирада «тескари» жойлашиши инобатга олинishi керак.

Агар ҳар хил турдаги кўрсаткичларга қийматлар берилса, албатта турга келтириш амалидан фойдаланиш керак:

```

int n=5;
float x=1.0;
int *pi=&n;
float *px=&x;
void *p;
int *r,*r1;
px=(float *)&n;
p=px;
r=(int *)p;
r1=pi;

```

Кўрсаткич турини воид турига келтириш амалда маънога эга эмас. Худди шундай, турлари бир хил бўлган кўрсаткичлар учун турни келтириш амалини бажаришга ҳожат йўқ.

Кўрсаткич устидан бажариладиган арифметик амалларда автоматик равишда турларнинг ўлчами ҳисобга олинади.

Арифметик амаллар фақат бир хил турдаги кўрсаткичлар устидан бажарилади ва улар асосан, массив тузилмаларига кўрсаткичлар устида бажарилади.

Инкремент амали кўрсаткични массивнинг кейинги элементига, декремент эса аксинча, битта олдинги элементининг адресига кўчиради. Бунда кўрсаткичнинг қиймати sizeof(<массив элементининг тури>) қийматига ўзгаради. Агар кўрсаткич k ўзгармас қийматга оширилса ёки камайтирилса, унинг қиймати k \* sizeof (<массив элементининг тури>) катталиқка ўзгаради.

Масалан:

```
short int *p=new short[5];
long * q=new long [5];
p++; // p qiymati 2 oshadi
q++; // q qiymati 4 ga oshadi
q+=3; // q qiymati 3*4=12 oshadi
```

Кўрсаткичларнинг айирмаси деб, улар айирмасининг тур ўлчамига бўлинишига айтилади. Кўрсаткичларни ўзаро кўшиш мумкин эмас.

#### 18.4. Адресни олиш амали

Адресни олиш қуйидагича эълон қилинади:

```
<тур> & <ном>;
```

Бу ерда <тур> - адреси олинган қийматнинг тури, <ном>- адрес олувчи ўзгарувчи номи. Ўртадаги ‘&’ белгисига *адресни олиш амали* дейилади.

Бу кўринишда эълон қилинган ўзгарувчи шу турдаги ўзгарувчининг синоними деб қаралади. Адресни олиш амали орқали битта ўзгарувчига ҳар хил ном билан мурожаат қилиш мумкин бўлади.

Мисол:

```
int kol;
int & pal=kol; // pal adres oluvchi o'zgaruvchi kol
                // o'zgaruvchisining alternativ nomi
const char & cr='\n'; // cr - o'zgaruvchiga murojaat
```

Адресни олиш амалини ишлатишда қуйидаги қоидаларга риоя қилиш керак: адрес олувчи ўзгарувчи функция параметри сифатида ишлатилган ёки эхтерн билан тавсифланган ёки синф майдонига мурожаат қилингандан ҳолатлардан ташқари барча ҳолатларда бошланғич қийматга эга бўлиши керак.

Адресни олиш амали асосан функцияларда адрес орқали узатилувчи параметрлар сифатида ишлатилади.

Адрес олувчи ўзгарувчининг кўрсаткичдан фарқи шундаки, у алоҳида хотирани эгалламайди ва фақат ўз қиймати бўлган ўзгарувчининг бошқа номи сифатида ишлатилади.

## 18.5. Кўрсаткичлар ва адрес олувчи ўзгарувчилар функция параметри сифатида

Функция прототипида ёки аниқланиш сарлавҳасида кўрсатилган параметрлар *формал параметрлар* дейилади, функция чақиришида кўрсатилган аргументларга *фактик параметрлар* дейилади.

Функция чақирилишида фактик параметрнинг тури мос ўриндаги формал параметр турига тўғри келмаса ёки шу турга келтиришнинг иложи бўлмаса компиляция хатоси рўй беради.

Фактик параметрларни функцияга икки хил усул билан узатиш мумкин: *қиймати ёки адреси* билан.

Функция чақирилишида аргумент қиймат билан узатилганда, аргумент ёки унинг ўрнидаги келган ифода қиймати ва бошқа аргументларнинг нусхаси (қийматлари) стек хотирасига ёзилади. Функция фақат шу нусхалар билан амал қилади, керак бўлса бу нусхаларга ўзгартиришлар қилиниши мумкин, лекин бу ўзгаришлар аргументнинг ўзига таъсир қилмайди, чунки функция ўз ишини тугатиши билан нусхалар ўчирилади (стек тозаланади).

Агар параметр адрес билан узатилса, стекка адрес нусхаси ёзилади ва худди шу адрес бўйича қийматлар ўқилади (ёзилади). Функция ўз ишини тугатгандан кейин шу адрес бўйича қилинган ўзгаришлар сақланиб қолинади ва бу қийматларни бошқа функциялар ишлатиши мумкин.

Аргумент қиймат билан узатилиши учун мос формал параметр сифатида ўзгарувчини тури ва номи ёзилади. Функция чақирилишида мос аргумент сифатида ўзгарувчининг номи ёки ифода бўлиши мумкин.

Фактик параметр адрес билан узатилганда унга мос келувчи формал параметрни икки хил усул билан ёзиш мумкин: *кўрсаткич орқали ёки адресни олувчи параметрлар орқали*. Кўрсаткич орқали ёзилганда формал параметр туридан кейин ‘\*’ белгиси ёзилади, мос аргументда эса ўзгарувчининг адреси (& амал орқали) ёки массив номи, ёки функция номи бўлиши мумкин. Адресни олиш амали орқали параметр узатишда формал параметрда туридан кейин ‘&’ белгиси ёзилади ва функция чақирилишида мос аргумент сифатида ўзгарувчи номи келади.

Мисол:

```
#include <iostream.h>
void f(int,int*,int &)
```

```

void main()
{
    int i=1,j=2,k=3;
    cout<<i<<" "<<j<<" "<<k;
    f(i,&j,k);
    cout<<i<<" "<<j<<" "<<k;
}
void f(int i,int *j,int &k)
{
    i++;
    (*j)++;
    k++;
    *j=i+k;
    k=*j+i;
}

```

Дастур ишлаши натижасида экранга қуйидаги қийматлар чоп қилинади:

```

1 2 3
1 6 8

```

Бу мисолда биринчи параметр *i* қиймат билан узатилади (“*int i*”). Унинг қиймати функция ичида ўзгаради, лекин ташқаридаги *i* ўзгарувчисининг қиймати ўзгармайди. Иккинчи параметрни кўрсаткич орқали адреси билан узатилиши талаб қилинади (“*int \*j*”), адресни узатиш учун ‘&’ – адресни олиш амали ишлатилган (“&*j*”). Функция танасида аргумент адресидан қиймат олиш учун ‘\*’- қиймат олиш амали қўлланилган. Учинчи параметрда мурожаат орқали (“&*k*”) аргументнинг адреси узатиш кўзда тутилган. Бу ҳолда функция чақирилишида мос аргумент ўрнида ўзгарувчи номи туради, функция ичида эса қиймат олиш амалини ишлатишнинг ҳожати йўқ. Функция ишлаш натижасидаги қийматларни аргументлар рўйхати орқали олиш қулай ва тушунарли усул ҳисобланади.

Агар функция ичида адрес билан узатиладиган параметр қиймати ўзгармасдан қолиши зарур бўлса, бу параметр сонет модификатор билан ёзилиши керак:

```

fun(int n,const char*str);

```

Агарда функцияни чақиришда аргументлар фақат номлари билан берилган бўлса, келишув бўйича массивлар ва функциялар адреси билан, қолган турдаги параметрлар қийматлари билан узатилган деб ҳисобланади.

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.- М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим мавзулари:

1. Адрес олиш амалининг функцияда ишлатиш афзалликлари.
2. Кўрсаткичли ўзгарувчиларни инициализация қилиш.
3. Кўрсаткичли ўзгарувчиларни инициализациялаш.
4. Функцияга кўрсаткичдан амалиётда фойдаланиш.

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include	include	препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
Кўрсаткич	indicator	қийматлари адрес бўлган ўзгарувчилар
Адрес	adress	ўзгарувчи хотирада жойлашадиган манзил
void		қиймати мавжуд бўлмаган тур
cout		экранга чиқариш оқими
cin		экрандан киритиш оқими

Инкремент		қийматни биттага ошириш
Функция		программа алоҳида бўлаги, асосий қисм томонидан чақириб ишлатилади
Ўзгарувчи	variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
Идентификатор		катта ва кичик лотин ҳарфлари, рақамлар ва таг чизик ('_') белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги

### Кейслар банки

**Кейс 1.** Функция тузилди ва функциянинг вазифаси массивдаги фибоначчи сонлари миқдорини аниқлаш. Функцияга кўрсаткич орқали мурожаат қилиб ишлатилди. Аммо дастур фибоначчи сонлари миқдорини нотўғри аниқлади. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир неча усулда дастур кодини тақдим этинг (индивидуал ҳолда).

#### Назорат саволлари:

1. Қийматлари адрес бўлган ўзгарувчиларга нима дейилади?
2. Кўрсаткич неча турда бўлади?
3. Функцияга кўрсаткичнинг ёзилиш синтаксиси қандай бўлади?
4. Объектга кўрсаткич эълони қандай бўлади?
5. void кўрсаткичининг муҳим афзалликлари нималардан иборат?
6. Динамик ўзгарувчилар деб нимага айтилади?
7. Кўрсаткичга бошланғич қиймат бериш қай тарзда амалга оширилади?
8. Кўрсаткич устида қандай амаллар бажарилиши мумкин?
9. Адрес олувчи ўзгарувчининг кўрсаткичдан фарқи нимадан иборат?
10. Формал ва параметрлар деб нимага айтилади?

## Тест саволлари:

1. Қуйидаги дастур кодининг натижасини топинг.

```
int x;  
int *p = &x;  
*p = 35;  
cout << x;
```

- a) 35
- b) 4
- c) 0
- d) 39

2. Қуйидаги дастур кодининг натижасини топинг.

```
int x;  
int y;  
int *p = &x;  
int *q = &y;  
*p = 35;  
*q = 98;  
*p = *q;  
cout << x << "  
" << y << endl;
```

- a) 98 98
- b) 35 98
- c) 98 35
- d) 35 35

3. Қуйидаги дастур кодининг натижасини топинг.

```
int x;  
int y=13;  
int *p = &x;  
int *q = &y;  
*p = 35;  
cout << x << " " << *q << endl;
```

- a) 35 13

- b)13 35
- c)35 0
- d)0 35

4. Қуйидаги дастур кодининг натижасини топинг.

```
int y=13;  
int *q = &y;  
y = 23;  
cout << *q << endl;
```

- a) 23
- b)14
- c) 0
- d) 0x0bbff

5. Қуйидаги дастур кодининг натижасини топинг.

```
int a = 0, b = 10;  
if (pow(b,a) == 1)  
    a = b * (++b);  
else  
    a = b * (--b);  
cout<<a;
```

- a) 121
- b) 111
- c) 100
- d) 81

6. Қуйидаги дастур кодининг натижасини топинг.

```
28 << 3 & 22 - 17 | 30 >> 2
```

- a) 7
- b) 8
- c) 9
- d) 10



7. Қуйидаги дастур кодининг натижасини топинг.

```
int a = 7, b = 8;  
if (++a == ++b) a += a;  
else if(a % 2 == 0) a += b * b;  
else if(a % 2 != 0) a += b;  
cout<<a;
```

- a) 89
- b) 98
- c) 91
- d) 90

8. Қуйидаги дастур кодининг натижасини топинг.

```
int a = 16, b = 5;  
if (sqrt(a) == b) a *= ++a;  
else a *= ++b;  
cout<<a;
```

- a) 96
- b) 80
- c) 36
- d) 81

9. Қуйидаги дастур кодининг натижасини топинг.

```
int a=6,b=4;  
while(a/b==1)  
{  
a++;  
if (b) a++;  
b--;  
a=b*3;  
}  
cout << a;
```

- a) 9
- b) 10
- c) 6
- d) 4

10. Қуйидаги дастур кодининг натижасини топинг.

```
int a = 0;
for (int i = 0; i < 50; i+=5)
{
    if (i / 10 >= 3)
        if (i % 10 != 5)
            a += i;
}
cout<<a;
```

- a) 70
- b) 50
- c) 10
- d) 150

## Маъруза 19. Динамик массивлар

### Режа:

1. new оператори.
2. delete оператори.
3. Динамик массивлар билан ишлаш.
4. new оператори ёрдамида динамик массивлар билан.

Таянч иборалар: *include, new, delete, alloc, calloc, malloc, free, массив, динамик массив, ўзгарувчи, идентификатор, кўрсаткич, адрес*

### 19.1. new оператори

Хотиранинг объектлар ўртасидан динамик тақсимланувчи соҳасидан жой ажратиш учун new оператори ишлатилади. new операторидан кейин хотирага жойлаштирилдиган объект типини кўрсатиш лозим. Бу объектни сақлаш учун талаб этиладиган хотира соҳаси ўлчовини аниқлаш учун керак бўлади. Масалан, `new unsigned short int` деб ёзиш орқали биз динамик тақсимланувчи хотирадан икки байт жой ажратамиз. Худди шунингдек, `new long` сатри орқали тўрт байт жой объектлар ўтрасида динамик тақсимланувчи соҳадан ажратилади.

new оператори натижа сифатида белгиланган хотира ячейкасининг адресини қайтаради. Бу адрес кўрсаткичга ўзлаштирилиши лозим. Масалан, `unsigned short` типигаги ўзгарувчи учун динамик соҳадан жой ажратиш учун қуйидаги дастур коди ёзилади:

```
unsigned short int *pPointer;  
pPointer = new unsigned short int;
```

Ёки худди шу амални битта сатрда ҳам ёзиш мумкин.

```
unsigned short int * pPoiner = new unsigned short int;
```

Иккала ҳолатда ҳам `pPointer` кўрсаткичи `unsigned short int` типигаги қийматни сақловчи динамик соҳа хотирасининг ячейкасини

кўрсатиб туради. Энда `pPointer` кўрсаткичини шу типдаги ихтиёрий ўзгарувчига кўрсаткич сифатида кўллаш мумкин. Ажратилган хотира соҳасига бирор бир қиймат жойлаштириш учун қуйидагича ёзув ёзилади:

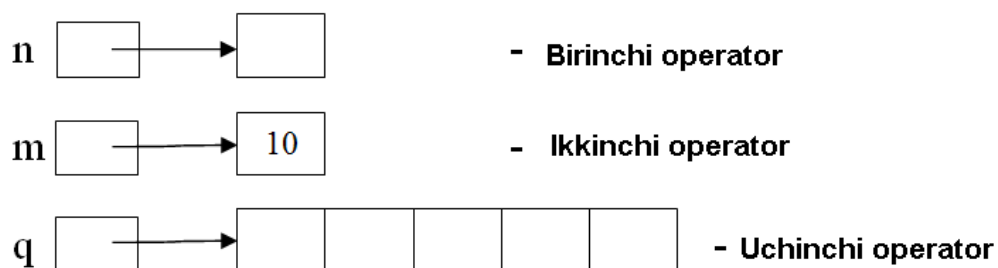
```
* pPointer = 72 ;
```

Бу сатр қуйидаги маънони англатади: «`pPointer` кўрсаткичида адреси сақланаётган хотирага 72 сонини ёзинг». Динамик хотира соҳаси албатта чегараланган бўлади. У тўлиб қолганда `new` оператори орқали хотирадан жой ажратишга уринсак хатолик юз беради.

Динамик хотирада `new` амали билан жой ажратиш ва уни адресини кўрсаткичга бериш:

```
int * n=new int;           // birinchi operator
int * m=new int(10);       // ikkinchi operator
int * q=new int[5];        // uchinchi operator
```

Биринчи операторда `new` амали ёрдамида динамик хотирада `int` учун этарли жой ажратиб олиниб, унинг адреси `n` кўрсаткичга юкланади. Кўрсаткичнинг ўзи учун жой компиляция вақтида ажратилади.



Динамик хотирадан жой ажратиш

Иккинчи операторда жой ажратишдан ташқари `m` адресига бошланғич қиймат - 10 сонини жойлаштиради.

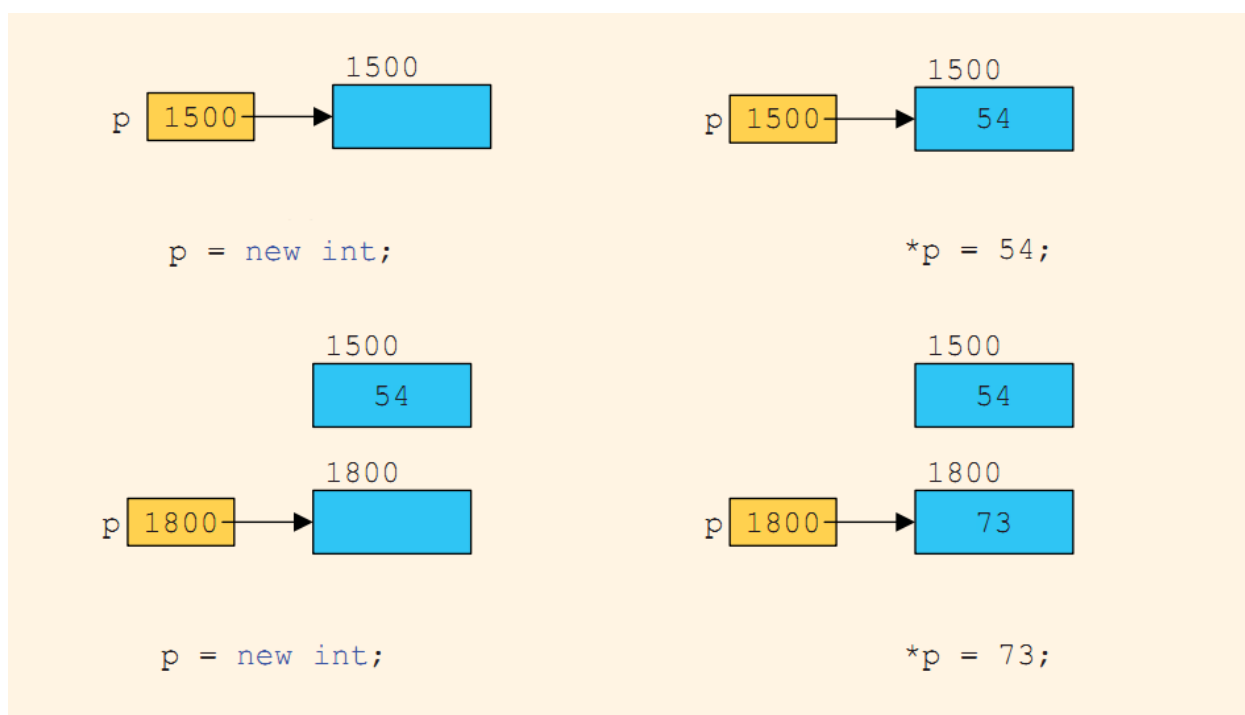
Учинчи операторда `int` туридаги 5 элемент учун жой ажратилган ва унинг бошланғич адреси `q` кўрсаткичга берилаяпти.

## 19.2. delete оператори

Агарда ўзгарувчи учун ажратилган хотира керак бўлмаса уни бўшатиш зарур. Бу ўзидан кейин кўрсаткич номи ёзиладиган `delete` оператори ёрдамида амалга оширилади. `delete` оператори кўрсаткич орқали аниқланган

хотира соҳасини бўшатади. Шунинг учун бу кўрсаткич эълон қилинган функциядан чиқишимиз билан кўрсаткич ҳам хотирадан ўчирилади. Лекин `new` оператори орқали бу кўрсаткичга динамик хотирадан ажратилган жой бўшатиб қўйилмайди. Натижада хотиранинг бу қисми киришга имконсиз бўлиб қолади. Дастурчилар бу ҳолатни хотиранинг сирқиб кетиши, ёки йўқолиши (утечка памяти) деб тавсифлайдилар. Бу тавсиф ҳақиқатга бутунлай мос келади, чунки дастур ишини якунлагунча хотирани бу қисмидан фойдаланиб бўлмайди.

```
int *p;
p = new int;
*p = 54;
p = new int;
*p = 73;
```



Хотира `new` амали билан ажратилган бўлса, у `delete` амали билан бўшатилиши керак. Юқоридаги динамик ўзгарувчилар билан боғланган хотира қуйидагича бўшатилади:

```
delete n; delete m; delete[]q;
```

Агарда хотира `new[]` амали билан ажратилган бўлса, уни бўшатиш учун

`new[]` амалини ўлчови кўрсатилмаган ҳолда қўллаш керак.

Хотира бўшатирилганлигига қарамадан кўрсаткични ўзини кейинчалик қайта ишлатиш мумкин.

```
delete pPointer;
```

Бунда кўрсаткич ўчирилмайди, балки унда сақланаётган адресдаги хотира соҳаси бўшатилади. Белгиланган хотирани бўшатилиши кўрсаткичга таъсир қилмайди, унга бошқа адресни ўзлаштириш ҳам мумкин. Қуйидаги листингда динамик ўзгарувчи учун қандай хотира ажратилиши, уни ишлатиш ва ажратилган хотирани бўшатишга оид мисол келтирилган.

```
#include <iostream>

using namespace std;

int main()
{
    int    local variable = 5;
    int * pLocal = & local variable;
    pHeap = 7;
    cout << "local variable:" << local variable << "\n";
    cout << " *pLocal: " << *pLocal << "\n";
    cout << " *pHeap: " << *pHeap << "\n";
    delete pHeap;
    pHeap = new int;
    *pHeap = 9;
    cout << "*pHeap:" << *pHeap << "\n";
    delete pHeap;
    return 0;
}
```

Дастур ишлаши натижасида экранга қуйидаги қийматлар чиқади:

```
local variable: 5
```

\*pLocal: 5

\*pHeap: 7

\*pHeap: 9

### 19.3. Динамик массивлар билан ишлаш

Статик массивларнинг камчиликлари шундаки, уларнинг ўлчамлари олдиндан маълум бўлиши керак, бундан ташқари бу ўлчамлар берилганларга ажратилган хотира сегментининг ўлчами билан чегараланган. Иккинчи томондан, этарлича катта ўлчамдаги массив эълон қилиб, конкрет масала эчилишида ажратилган хотира тўлиқ ишлатилмаслиги мумкин. Бу камчиликлар динамик массивлардан фойдаланиш орқали бартараф этилади, чунки улар дастур ишлаши жараёнида керак бўлган ўлчамдаги массивларни яратиш ва зарурат қолмаганда йўқотиш имкониятини беради.

Динамик массивларга хотира ажратиш учун `malloc()`, `calloc()` функцияларидан ёки `new` операторидан фойдаланиш мумкин. Динамик объектга ажратилган хотирани бўшатиш учун `free()` функцияси ёки `delete` оператори ишлатилади.

Юқорида қайд қилинган функциялар «`alloc.h`» кутубхонасида жойлашган.

`malloc()` функциясининг синтаксиси

```
void * malloc(size_t size);
```

Кўринишида бўлиб, у хотиранинг уюм қисмидан сизе байт ўлчамидаги узлуксиз соҳани ажратади. Агар хотира ажратиш муваффақиятли бўлса, `malloc()` функцияси ажратилган соҳанинг бошланиш адресини қайтаради. Талаб қилинган хотирани ажратиш муваффақиятсиз бўлса, функция `NULL` қийматини қайтаради.

Синтаксисдан кўришиб турибдики, функция `void` туридаги қиймат қайтаради. Амалда эса конкрет турдаги объект учун хотира ажратиш зарур бўлади. Бунинг учун воид турини конкрет турга келтириш технологиясидан фойдаланилади. Масалан, бутун турдаги узунлиги 3 га тенг массивга жой ажратишни қуйидагича амалга ошириш мумкин:

```
int * pInt=(int*)malloc(3*sizeof(int));
```

malloc() функцияси malloc() функциясидан фаркли равишда массив учун жой ажратишдан ташқари массив элементларини 0 қиймати билан инициализатсия қилади. Бу функция синтаксиси

```
void * calloc(size_t num, size_t size);
```

қўринишда бўлиб, нум параметри ажратилган соҳада нечта элемент борлигини, сизе ҳар бир элемент ўлчамини билдиради.

free() хотирани бўшатиш функцияси ўчириладиган хотира бўлагига кўрсаткич бўлган ягона параметрга эга бўлади:

```
void free(void * block);
```

free() функцияси параметрининг void турида бўлиши ихтиёрий турдаги хотира бўлагини ўчириш имконини беради.

Қуйидаги дастурда 10 та бутун сондан иборат динамик массив яратиш, унга қиймат бериш ва ўчириш амаллари бажарилган.

```
#include <iostream>
using namespace std;
#include <alloc.h>
int main()
{
    int * pVector;
    if ((pVector=(int*)malloc(10*sizeof(int)))==NULL)
    {
        cout<<"Xotira etarli emas!!!";
        return 1;
    }
    // ajratilgan xotira sohasini to'ldirish
    for(int i=0;i<10;i++) *(pVector+i)=i;
    // vektor elementlarini chop etish
```



```

for(int i=0; i<10; i++) cout<<*(pVector+i)<<endl;
// ajratilgan xotira bo'lagini qaytarish (o'chirish)
free(pVector);
return 0;
}

```

Кейинги дастурда  $n \times n$  ўлчамли ҳақиқий сонлар массивининг бош диагоналидан юқорида жойлашган элементлар йиғиндисини ҳисоблаш масаласи ечилган.

```

#include <iostream>
using namespace std;
#include <alloc.h>
int main()
{
    int n;
    float * pMatr, s=0;
    cout<<"A(n,n): n=";
    cin>>n;
    if((pMatr=(float*)malloc(n*n*sizeof(float)))==NULL)
    {
        cout<<"Xotira etarli emas!!!";
        return 1;
    }
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)cin>>*(pMatr+i*n+j);
    for(int i=0;i<n;i++)
        for(int j=i+1;j<n;j++)s+=*(pMatr+i*n+j);
    cout<<"Matritsa bosh diagonalidan yuqoridagi ";

```

```
cout<<"elementlar yig`indisi S="<<s<<endl;
return 0;
}
```

#### **19.4. new operatori ёрдамида динамик массивлар билан ишлаш**

new operatori ёрдамида, массивга хотира ажратишда объект туридан кейин квадрат кавс ичида объектлар сони кўрсатилади. Масалан, бутун турдаги 10 та сондан иборат массивга жой ажратиш учун

```
int * pVector;
pVector=new int[10];
```

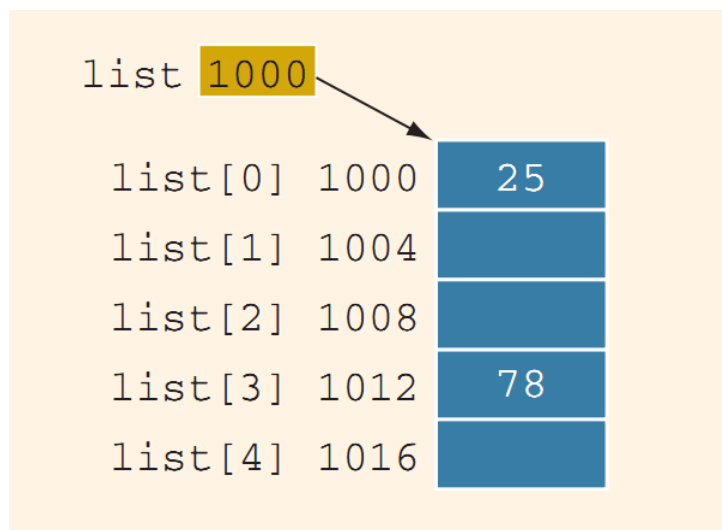
ифодаси ёзилиши керак. Бунга қарама-қарши равишда, бу усулда ажратилган хотирани бўшатиш учун

```
delete [] pVector;
```

кўрсатмасини бериш керак бўлади.

Қиймат бериш индекси орқали амалга оширилади.

```
int * list;
list=new int[5];
list[0] = 25;
list[3] = 78;
for (j = 0; j < 10; j++)
    list[j] = 0;
```



Икки ўлчамли динамик массивни ташкил қилиш учун

```
int **a;
```

кўринишидаги «кўрсаткичга кўрсаткич» ишлатилади.

Бошда массив сатрлари сонига қараб кўрсаткичлар массивига динамик хотирадан жой ажратиш керак:

```
a=new int *[m] // bu erda m massiv satrlari soni
```

Кейин, ҳар бир сатр учун такрорлаш оператори ёрдамида хотира ажратиш ва уларнинг бошланғич адресларини а массив элементларига жойлаштириш зарур бўлади:

```
for(int i=0;i<m;i++)
```

```
    a[i]=new int[n]; //n ustunlar soni
```

Шуни қайд этиш керакки, динамик массивнинг ҳар бир сатри хотиранинг турли жойларида жойлашиши мумкин.

Икки ўлчамли массивни ўчиришда олдин массивнинг ҳар бир элементи (сатри), сўнгра массивнинг ўзи йўқотилади:

```
for(i=0; i<m; i++)
```

```
    delete[] a[i];
```

```
delete[] a;
```

Матрицани векторга кўпайтириш масаласи учун динамик массивлардан фойдаланишга мисол:

```

void main ()
{
int n,m;
int i,j; float s;
cout<<"\n n="; cin>>n; // matritsa satrlari soni
cout<<"\n m="; cin>>m; // matritsa ustunlari soni
float *b=new float[m];
float *c=new float[n];
// ko'rsatkichlar massiviga xotira ajratish
float **a=new float *[n] ;
for(i=0;i<n;i++) // har bir satr uchun
a[i]=new float[m]; //dinamik xotira ajratish
for(j=0;j<m;j++)
    cin>>b[j];
for(i=0;i<n;i++)
    for(j=0;j<m;j++)
        cin>>a[i][j];
for(i=0;i<n;i++)
{
    for(j=0,s=0;j<m;j++)
        s+=a[i][j]*b[j];
    c[i]=s;
}
for(i=0;i<n;i++)
    cout<<"\t c["<<i<<"]="<<c[i];
delete[]b;
delete[]c;

```

```

for (i=0;i<n;i++)
    delete[]a[i];
delete[]a;
return;
}

```

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. Мадрахимов Ш.Ф., Гайназаров С.М. С++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим мавзулари

1. Динамик массив элементларининг хотирадаги кўриниши.
2. Динамик массив элементларини тасодифий сонлар билан тўлдириш.
3. С++/CLI кўринишидаги динамик массивлар.
4. Динамик массив устида амаллар бажариш.

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
Дастурлаш тиллари	Programming languages	дастур таъминотини яратиш жараёнини осонлаштириш учун яратилган тиллар
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
cout		экранга чиқариш оқими
кенгайтма	Extension	файлларнинг турли дастурларга тегишлилигини аниқловчи файл кўринишининг қисми

Компиляция	Compilation	бажарилувчи файл хосил бўлиш жараёни
Идентификатор	Identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизиқ (‘_’) белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги
new		хотирадан янги жой ажратиш оператори
delete		хотирадан ажратилган жойни тозалаш оператори
Кўрсаткич	Pointer	қийматлари адрес бўлган ўзгарувчилар
Адрес	Address	ўзгарувчи хотирада жойлашадиган манзил
cout		экранга чиқариш оқими.
cin		экрандан киритиш оқими
Ўзгарувчи	Variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги

### Кейслар банки

**Кейс 1.** Динамик массивлар билан ишлайдиган дастур тузилди. Дастур вазифаси массив элементларини циклик равишда чапга n та суриш. Дастур ишлаши натижасида хатолик келиб чиқди. Яъни илова хатолик ҳақида хабар берди.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

## Назорат саволлари

1. Массивлар нима мақсадда ишлатилади?
2. new оператори натижа сифатида нимани қайтаради?
3. Динамик хотирада new амали билан жой ажратиш?
4. Керак бўлмаган хотирани қайси оператор ёрдамида бўшатиш мумкин?
5. Динамик массив билан статик массивнинг фарқини айтиб беринг.
6. Қайси операторлар ёрдамида динамик массив билан ишлаш имкони туғилади?
7. «alloc.h» кутубхонасида аниқланган функциялар ва уларнинг вазифаларини айтиб беринг.
8. Динамик массив элементлари миқдорини қандай кўрсатиш мумкин?
9. Бир ўлчамли динамик массив эълон қилиниши ва қиймат олишига мисол келтиринг.
10. Кўп ўлчамли динамик массив эълон қилиниши ва қиймат олишига мисол келтиринг.

## Тест саволлари

1. C++ тилида динамик массивларга хотирадан жой ажратиш учун мўлжалланган функциялар ва операторлар тўғри кўрсатилган жавобни танланг.
  - a) malloc(), calloc() функциялари ва new оператори
  - b) int(), float() функциялари ва new operator
  - c) free() функцияси ва delete оператори
  - d) malloc() функцияси ва new, delete операторлари
2. C++ тилида динамик объект яратувчи операторни танланг.
  - a) new
  - b) sizeof
  - c) delete
  - d) casting

3. C++ тилида динамик массив учун ажратилган жойни ўчириш функциясини кўрсатинг.

- a) delete
- b) sizeof
- c) new
- d) casting

4. calloc() функцияси қайси кутубхонада жойлашган?

- a) calloc.h
- b) dynamic.h
- c) massiv.h
- d) new.h

5. C++ тилида malloc оператори ёрдамида динамик массив учун хотирадан ажратилган жойни қандай ўчирилади?

- a) free(pVector);
- b) delete pvector
- c) free pVector
- d) detele(pVector)

6. C++ тилида calloc оператори синтаксисини кўрсатинг.

- a) void \* calloc(size\_t num, size\_t size);
- b) void \* calloc( size\_t size);
- c) void \* calloc(size\_t num,char c);
- d) void \* calloc(size\_t num, size\_t size, type);

7. C++ тилида malloc оператори ёрдамида динамик массив учун n та бутун турда жой ажратишни кўрсатинг.

- a) int \* pInt=(int\*)malloc(n\*sizeof(int));
- b) int \* pInt=malloc(n\*sizeof(int));
- c) int \* pInt=(int\*)malloc(n\*sizeof(n));
- d) int \* pInt=( \*)malloc(n\*sizeof(n));

8. Қиймат қайтармайдиган тур тўғри берилган жавобни кўрсатинг.

- a) void



- b) double
- c) int
- d) char

9. Хотирадан жой ажратиш функцияси тўғри берилган жавобни кўрсатинг.

- a) malloc()
- b) delete()
- c) int()
- d) free()

10. Объектга ажратилган хотирани бўшатиш учун қайси функция ишлатилади?

- a) free()
- b) calloc()
- c) int()
- d) delete()

## Маъруза 20-21. Сатрлар. Сатр устида амаллар. Сатр функциялари

### Режа:

1. ASCIIZ-сатрлар
2. ASCIIZ-сатрлар узунлигини аниқлаш функциялари
3. ASCIIZ-сатрларни нусхалаш
4. ASCIIZ-сатрларни улаш
5. ASCIIZ-сатрларда излаш функциялари
6. string туридаги сатрлар
7. Сатр қисмини бошқа сатрга нусхалаш функцияси
8. Сатр қисмини ўчириш ва алмаштириш функциялари
9. Сатр қисмини излаш ва солиштириш функциялари

Таянч иборалар: *сатр*, *ASCIIZ-сатрлар*, *sizeof()* функцияси, *getline()* функцияси, *string*, *сатр* қисми

### 20.1. ASCIIZ-сатрлар

Стандарт C++ тили икки хилдаги белгилар мажмуасини қўллаб-қувватлайди. Биринчи тоифага, анъанавий, «тор» белгилар деб номланувчи 8-битли белгилар мажмуаси киради, иккинчисига 16-битли «кенг» белгилар киради. Тил кутубхонасида ҳар бир гуруҳ белгилари учун махсус функциялар тўплами аниқланган.

C++ тилида сатр учун махсус тур аниқланмаган. Сатр чар туридаги белгилар массиви сифатида қаралади ва бу белгилар кетма-кетлиги *сатр терминатори* деб номланувчи 0 кодли белги билан тугайди ('\0'). Одатда, нол-терминатор билан тугайдиган сатрларни *ASCIIZ-сатрлар* дейилади.

Қуйидаги жадвалда C++ тилида белги сифатида ишлатилиши мумкин бўлган ўзгармаслар тўплами келтирилган.

#### C++ тилидаги белги ўзгармаслар

Белгилар синфлари	Белги ўзгармаслар
Катта ҳарфлар	'A' ... 'Z', 'A' ... 'YA'
Кичик ҳарфлар	'a' ... 'z', 'a' ... 'ya'
Рақамлар	'0' ... '9'

Бўш жой	gorizontal tabulyasiya (ASCII kodi 9), satrni o'tkazish (ASCII kodi 10), vertikal tabulyasiya (ASCII kodi 11), formani o'tkazish (ASCII kodi 12), karetkani qaytarish (ASCII kodi 13)
Пунктуатсия белгилари (ажратувчилар)	! " # \$ % & ' ( ) * + - , . / : ; < = > ? @ [ \ ] ^ _ {   } ~
Бошқарув белгилари	ASCII kodi 0...1Fh oraliq'ida va 7Fh bo'lgan belgilar
Пробел	ASCII kodi 32 bo'lgan belgi
Ўн олтилик рақамлар	'0'...'9', 'A'...'F', 'a'...'f'

Сатр массиви эълон қилинишида, сатр охирига терминатор қўйилиши ва натижада сатрга қўшимча битта байт бўлишини инобатга олиниши керак:

```
char satr[10];
```

Ушбу эълонда сатр сатри учун жами 10 байт ажратилади - 9 сатр ҳосил қилувчи белгилар учун ва 1 байт терминатор учун.

Сатр ўзгарувчилар эълон қилинишида бошланғич қийматларни қабул қилиши мумкин. Бу ҳолда компилятор автоматик равишда сатр узунлиги ҳисоблайди ва сатр охирига терминаторни қўшиб қўяди:

```
char Hafta_kuni[]="Juma";
```

Ушбу эълон қуйидаги эълон билан эквивалент:

```
char Hafta_kuni[]={ 'J', 'u', 'm', 'a', '\0' };
```

Сатр қийматини ўқишда оқимли ўқиш оператори ">>" ўрнига `getline()` функциясини ишлатган маъқул ҳисобланади, чунки оқимли ўқишда пробеллар инкор қилинади (гарчи улар сатр белгиси ҳисобланса ҳам) ва ўқиладиган белгилар кетма-кетлиги сатрдан «ошиб» кетганда ҳам белгиларни киритиш давом этиши мумкин. Натижада сатр ўзига ажратилган ўлчамдан ортиқ белгиларни «қабул» қилади. Шу сабабли, `getline()` функцияси иккита параметрга эга бўлиб, биринчи параметр ўқиш амалга ошириладиган сатрга кўрсаткич, иккинчи параметрда эса ўқилиши керак бўлган белгилар сони кўрсатилади. Сатрни `getline()` функцияси орқали ўқишга мисол кўрайлик:

```
#include <iostream>
using namespace std;
int main()
{
    char satr[6];
```

```

cout<<"Satrni kiriting: "<<'\\n';
cin.getline(satr,6);
cout<<"Siz kiritgan satr: "<<satr;
return 0;
}

```

Дастурда ишлатилган сатр сатри 5 та белгини қабул қилиши мумкин, ортикчалари ташлаб юборилади. getline() функциясига мурожаатда иккинчи параметр қиймати ўқиладиган сатр узунлигидан катта бўлмаслиги керак.

Сатр билан ишлайдиган функцияларнинг аксарияти «cstring» (string.h) кутубхонасида жамланган. Нисбатан кўп ишлатиладиган функцияларнинг тавсифини келтирамиз.

## 20.2. ASCIIZ-сатрлар узунлигини аниқлаш функциялари

Сатрлар билан ишлашда, аксарият ҳолларда сатр узунлигини билиш зарур бўлади. Бунинг учун « string.h » кутубхонасида strlen() функцияси аниқланган бўлиб, унинг синтаксиси қуйидагича бўлади:

```
size_t strlen(const char* string)
```

Бу функция узунлиги ҳисобланиши керак бўлган сатр бошига кўрсаткич бўлган ягона параметрга эга ва у натижа сифатида ишорасиз бутун сонни қайтаради. strlen() функцияси сатрнинг реал узунлигидан битта кам қиймат қайтаради, яъни нол-терминатор ўрни ҳисобга олинмайди.

Худди шу мақсадда sizeof() функциясидан ҳам фойдаланиш мумкин ва у strlen() функциясидан фарқли равишда сатрнинг реал узунлигини қайтаради. Қуйида келтирилган мисолда сатр узунлигини ҳисоблашнинг ҳар иккита варианты келтирилган:

```

#include <iostream>
using namespace std;
#include <string.h>
int main()
{
char Str[]="1234567890";
cout <<"strlen(Str)="<<strlen(Str)<<endl;
cout <<"sizeof(Str)="<<sizeof(Str)<<endl;

```

```
    return 0;
}
```

Дастур ишлаши натижасида экранга

```
strlen(Str)=10
```

```
sizeof(Str)=11
```

хабарлари чиқади.

Одатда `sizeof()` функциясидан `getline()` функциясининг иккинчи аргументи сифати ишлатилади ва сатр узунлигини яққол кўрсатмаслик имконини беради:

```
cin.getline(Satr, sizeof(Satr));
```

**Масала.** Фақат лотин ҳарфларидан ташкил топган сатр берилган. Ундаги ҳар хил ҳарфлар миқдори аниқлансин.

```
int main()
{
    const int n=80;
    char Satr[n];
    cout<<"Satrni kiriting:";
    cin.getline(Satr,sizeof(Satr));
    float s=0;
    int k;
    for(int i=0;i<strlen(Satr); i++)
        if(Satr[i]!=' ')
        {
            k=0;
            for(int j=0;j<strlen(Satr); j++)
                if(Satr[i]==Satr[j]||abs(Satr[i]-Satr[j])==32)
                    k++;
            s+=1./k;
        }
    cout<<"Satrdagi turli harflar miqdori: "<<(int)s;
    return 0;
}
```

Дастурда сатр учун 80 узунлигидаги Сатр белгилар массиви эълон қилинган ва унинг қиймати клавиатурадан киритилади. Масала қуйидагича эчилади. Ичма-ич жойлашган такрорлаш оператори ёрдамида `Satr` массивининг ҳар бир элементи - `Satr[i]` массивнинг барча элементлари - `Satr[j]` билан устма-уст тушиши ёки улар бир-биридан 32 сонига фарк қилиши (катта ва кичик лотин ҳарфларининг кодлари ўртасидаги фарк) ҳолатлари к ўзгарувчисида саналади ва с умумий йиғиндига  $1/k$  қиймати билан қўшилади. Дастур охирида с қиймати бутун турга айлантирилган ҳолда чоп этилади. Сатрдаги сўзларни бир-биридан ажратувчи пробел белгиси чеклаб ўтилади.

Дастурга

```
Satrdagi turli harflar miqdori
```

```
сатри киритилса, экранга жавоб тариқасида
```

```
Satrdagi turli belgilar miqdori: 13
```

```
сатри чоп этилади.
```

### 20.3. ASCIIZ-сатрларни нусхалаш

Сатр қийматини бирдан иккинчисига нусхалаш мумкин. Бу мақсадда бир қатор стандарт функциялар аниқланган бўлиб, уларнинг айримларининг тавсифларини келтираамиз.

`strcpy()` функцияси прототипи

```
char* strcpy(char* str1, const char* str2)
```

кўринишга эга ва бу функция `str2` сатрдаги белгиларни `str1` сатрга байтма-байт нусхалайди. Нусхалаш `str2` кўрсатиб турган сатрдаги нол-терминал учрагунча давом этади. Шу сабабли, `str2` сатр узунлиги `str1` сатр узунлигидан катта эмаслигига ишонч ҳосил қилиш керак, акс ҳолда берилган соҳасида (сегментда) `str1` сатрдан кейин жойлашган берилганлар «устига» `str2` сатрнинг «ортиб қолган» қисми ёзилиши мумкин.

Навбатдаги дастур қисми “Сатрни нусхалаш!” сатрини `Str` сатрга нусхалайди:

```
char Str[20];
```

```
strcpy(Str, "Satrni nusxalash!");
```

Зарур бўлганда сатрнинг қайсидир жойидан бошлаб, охиригача

нусхалаш мумкин. Масалан, “Сатрни нусхалаш!” сатрини 8-белгисидан бошлаб нусха олиш зарур бўлса, уни қуйидагича ечиш мумкин:

```
#include <iostream>
using namespace std;
#include <string.h>
int main()
{
    char Str1[20]="Satrni nusxalash!",Str2[20];
    char* kursatkich=Str1;
    kursatkich+=7;
    strcpy(Str2,kursatkich);
    cout<<Str2<<endl;
    return 0;
}
```

strcpy() функциясининг strncpy() функциясидан фарқли жойи шундаки, унда бир сатрдан иккинчисига нусхаланадиган белгилар сони кўрсатилади. Унинг прототипи қуйидаги кўринишга эга:

```
char* strncpy(char*str1, const char*str2,size_t num);
```

Агар str1 сатр узунлиги str2 сатр узунлигидан кичик бўлса, ортиқча белгилар «кесиб» ташланади. strncpy() функцияси ишлатилишига мисол кўрайлик:

```
#include <iostream>
using namespace std;
#include <string.h>
int main()
{
    char Uzun_str[]="01234567890123456789";
    char Qisqa_str[]="ABCDEF";
    strncpy(Qisqa_str,Uzun_str,4);
    cout <<"Uzun_str= "<<Uzun_str<<endl;
    cout<<"Qisqa_str="<<Qisqa_str<<endl;
    return 0;
}
```

Дастурда Узун\_стр сатри бошидан 4 белги Қисқа\_стр сатрига, унинг олдинги қийматлари устига жойланади ва натижада экранга

```
01234567890123456789
```

```
0123EF
```

сатрлар чоп этилади.

strdup() функциясига ягона параметр сифатида сатр-манбага кўрсаткич узатилади. Функция, сатрга мос хотирадан жой ажратади, унга сатрни нусхалайди ва юзага келган сатр нусха адресини жавоб сифатида қайтаради. strdup() функция синтаксиси:

```
char* strdup(const char* source)
```

Қуйидаги дастур бўлагида satr1 сатрининг нусхаси хотиранинг satr2 кўрсатган жойида пайдо бўлади:

```
char* satr1="Satr nusxasini olish."; char* satr2;  
satr2=strdup(satr1);
```

#### 20.4. ASCIIZ-сатрларни улаш

Сатрларни улаш (конкатенация) амали янги сатрларни ҳосил қилишда кенг қўлланилади. Бу мақсадда « string.h » кутубхонасида strcat() ва strncat() функциялари аниқланган.

strcat() функцияси синтаксиси қуйидаги кўринишга эга:

```
char* strcat(char* str1, const char* str2)
```

Функция ишлаши натижасида str2 сатр, функция қайтарувчи сатр - str1 сатр охирига уланади. Функцияни чақиришдан олдин str1 сатр узунлиги, унга str2 сатри уланиши учун этарли бўлиши ҳисобга олинган бўлиши керак.

Қуйида келтирилган амаллар кетма-кетлигининг бажарилиши натижасида сатр сатрига қўшимча сатр уланиши кўрсатилган:

```
char satr[80];  
strcpy(satr,"Bu satrga ");  
strcat(satr,"satr osti ulandi.");
```

Амаллар кетма-кетлигини бажарилиши натижасида сатр кўрсатаётган жойда "Бу сатрга сатр ости уланди." сатри пайдо бўлади.

strncat() функцияси strcat() функциядан фарқли равишда str1 сатрга str2 сатрнинг кўрсатилган узунликдаги сатр қисмини улайди. Уланадиган сатр



қисми узунлиги функциянинг учинчи параметри сифатида берилади.

Функция синтаксиси

```
char* strncat(char* str1,const char* str2,size_t num)
```

Пастда келтирилган дастур бўлагида стр1 сатрга стр2 сатрнинг бошланғич 10 та белгидан иборат сатр қисмини улайди:

```
char satr1[80]="Dasturlash tillariga misol bu-";
```

```
char satr2[80]="C++,Pascal,Basic";
```

```
strncat(satr1,satr2,10);
```

```
cout << satr1;
```

Амаллар бажарилиши натижасида экранга

```
Dasturlash tillariga misol bu-C++,Pascal
```

сатри чоп этилади.

## 20.5. ASCIIZ-сатрларда излаш функциялари

Сатрлар билан ишлашда ундаги бирорта белгини излаш учун «string.h» кутубхонасида бир қатор стандарт функциялар мавжуд.

Бирорта белгини берилган сатрда бор ёки йўқлигини аниқлаб берувчи strchr() функциясининг прототипи

```
char* strchr(const char* string, int c);
```

Кўринишида бўлиб, у с белгинининг string сатрида излайди. Агар излаш мувофақиятли бўлса, функция шу белгининг сатрдаги ўрнини (адресини) функция натижаси сифатида қайтаради, акс ҳолда, яъни белги сатрда учрамаса функция NULL қийматини қайтаради. Белгини излаш сатр бошидан бошланади.

Қуйида келтирилган дастур бўлаги белгини сатрдан излаш билан боғлиқ.

```
char satr[]="0123456789";
```

```
char* pSatr;
```

```
pSatr=strchr(satr,'6');
```

Дастур ишлаши натижасида pSatr кўрсаткичи сатр сатрининг '6' белгиси жойлашган ўрни адресини кўрсатади.

strchr() функцияси берилган белгини берилган сатр охиридан бошлаб излайди. Агар излаш мувофақиятли бўлса, белгини сатрга охириги

киришининг ўрнини қайтаради, акс ҳолда NULL.

Мисол учун

```
char satr[]="0123456789101112";
```

```
char* pSatr;
```

```
pSatr=strrchr(satr,'0');
```

амалларини бажарилишида pSatr кўрсаткичи satr сатрининг "01112" сатр қисмининг бошланишига кўрсатади.

strspn() функцияси иккита сатрни белгиларни солиштиради. Функция куйидаги

```
size_t strspn(const char* str1, const char* str2);
```

кўринишга эга бўлиб, у str1 сатрдаги str2 сатрга кирувчи бирорта белгини излайди ва агар бундай элемент топилса, унинг индекси функция қиймати сифатида қайтарилади, акс ҳолда функция сатр узунлигидан битта ортик қийматни қайтаради.

Мисол:

```
char satr1[]="0123ab6789012345678";
```

```
char satr2[]="a32156789012345678";
```

```
int farqli_belgi;
```

```
farqli_belgi=strspn(satr1,satr2);
```

```
cout<<"Satr1 satridagi Satr2 satrga kirmaydigan\
```

```
birinchi belgi indexsi = "<<farqli_belgi;
```

```
cout<<"va u '"<<satr1[farqli_belgi]<<"' belgisi.";
```

амаллар бажарилиши натижасида экранга

```
Satrlardagi mos tushmagan belgi indexsi = 5
```

сатри чоп этилади.

strcspn () функциясининг прототипи

```
size_t strcspn(const char* str1, const char* str2);
```

кўринишида бўлиб, у str1 ва str2 сатрларни солиштиради ва str1 сатрининг str2 сатрига кирган биринчи белгини индексини қайтаради. Масалан,

```
char satr[]="Birinchi satr";
```

```
int index;
```

```
index=strcspn(satr,"sanoq tizimi");
```

амаллари бажарилгандан кейин `index` ўзгарувчиси 1 қийматини қабул қилади, чунки биринчи сатрнинг биринчи ўриндаги белгиси иккинчи сатрда учрайди.

`strpbrk ()` функциясининг прототипи

```
char* strpbrk(const char* str1, const char* str2);
```

Кўринишга эга бўлиб, у `str1` сатрдаги `str2` сатрга кирувчи бирорта белгини излайди ва агар бундай элемент топилса, унинг адреси функция қиймати сифатида қайтарилади, акс ҳолда функция `NULL` қиймати қайтаради. Қуйидаги мисол функцияни қандай ишлашини кўрсатади.

```
char satr1[]="0123456789ABCDEF";
```

```
char satr2[]="ZXYabcdefABC";
```

```
char* element;
```

```
element = strpbrk(satr1,satr2);
```

```
cout<<element<<'\n';
```

Дастур ишлаши натижасида экранга `str1` сатрининг

```
ABCDEF
```

сатр остиси чоп этилади.

Сатрлар билан ишлашда бир сатрда иккинчи бир сатрнинг (ёки унинг бирор қисмини) тўлиқ киришини аниқлаш билан боғлиқ масалалар нисбатан кўп учрайди. Масалан, матн таҳрирларидаги сатрдаги бирорта сатр қисмини иккинчи сатр қисми билан алмаштириш масаласини мисол келтириш мумкин (юқорида худди шундай масала учун дастур келтирилган). Стандарт «`string.h`» кутубхонаси бу тоифадаги масалалар учун бир нечта функцияларни таклиф этади.

`strstr ()` функцияси қуйидагича эълон қилинади:

```
char* strstr(const char* str, const char* substr);
```

Бу функция `str` сатрига `substr` сатр қисми кириши текширади, агар `substr` сатр қисми `str` сатрига тўлиқ кириши мавжуд бўлса, сатрнинг чап томонидан биринчи киришдаги биринчи белгининг адреси жавоб тариқасида қайтарилади, акс ҳолда функция `NULL` қийматини қайтаради.

Қуйидаги мисол `strstr()` функциясини ишлатишни кўрсатади.

```
char satr1[]=
```

```
"Satrdan satr ostisi izlanmoqda, satr ostisi mavjud";
```

```
char satr2[]="satr ostisi";
char* satr_osti;
satr_osti=strstr(satr1,satr2);
cout<<satr_osti<<'\n';
```

Дастур буйруқлари бажарилиши натижасида экранга satr ostisi izlanmoqda, satr ostisi mavjud сатри чоп этилади.

Кейинги дастур бўлагида сатрда бошқа бир сатр қисми мавжуд ёки йўқлигини назорат қилиш ҳолати кўрсатилган:

```
char Ismlar[]=
"Alisher,Farxod, Munisa, Erkin, Akmal, Nodira";
char Ism[10];
char* Satrdagi_ism;
cout<< "Ismni kiriting: "; cin>>Ism;
Satrdagi_ism = strstr(Ismlar,Ism);
cout<< "Bunaqa ism ru\`yxatda ";
if (Satrdagi_ism==NULL) cout << "yo\`q ." <<'\n';
else cout<< "bor ." <<'\n';
```

Дастурда фойдаланувчидан сатр қисми сифатида бирорта номни киритиш талаб қилинади ва бу қиймат Исм сатрига ўқилади. Киритилган исм дастурда аниқланган рўйхатда (Исmlар сатрида) бор ёки йўқлиги аниқланади ва хабар берилади.

strtok () функциясининг синтаксиси

```
char* strtok(char* str, const char* delim);
```

кўринишда бўлиб, у стр сатрида делим сатр-рўйхатида берилган ажратувчилар оралиғига олинган сатр қисmlарни ажратиб олиш имконини беради. Функция биринчи сатрда иккинчи сатр-рўйхатдаги ажратувчини учратса, ундан кейин нол-терминаторни қўйиш орқали стр сатрни иккига ажратади. Сатрнинг иккинчи бўлагидан ажратувчилар билан «ўраб олинган» сатр қисmlари топиш учун функцияни кейинги чақирилишида биринчи параметр ўрнига NULL қийматини қўйиш керак бўлади. Қуйидаги мисолда сатрни бўлақларга ажратиш масаласи қаралган:

```

#include <iostream>
using namespace std;
#include <string.h>
int main()
{
    char Ismlar[]=
    "Alisher,Farxod Munisa, Erkin? Akmal0, Nodira";
    char Ajratuvchi[]=" ,!?.0123456789";
    char* Satrdagi_ism;
    Satrdagi_ism=strtok(Ismlar,Ajratuvchi);
    if(Satrdagi_ism) cout<<Satrdagi_ism<<'\n';
    while(Satrdagi_ism)
    {
        Satrdagi_ism=strtok(NULL,Ajratuvchi);
        if(Satrdagi_ism) cout<<Satrdagi_ism<<'\n';
    }
    return 0;
}

```

Дастур ишлаши натижасида экранга Исmlар сатридаги ‘ ’ (пробел), ‘,’ (вергул), ‘?’ (сўроқ белгиси) ва ‘0’ (рақам) билан ажратилган сатр қисmlари - исmlар чоп қилинади:

```

Alisher
Farxod
Munisa
Erkin
Akmal
Nodira

```

## 20.6. string туридаги сатрлар

C++ тилида стандарт сатр турига қўшимча сифатида string тури киритилган ва у string синфи кўринишида амалга оширилган. Бу турдаги сатр учун ‘\0’ белгиси тугаш белгиси ҳисобланмайди ва у оддийгина белгилар массиви сифатида қаралади. string турида сатрлар узунлигининг

бажариладиган амаллар натижасида динамик равишда ўзгариб туриши, унинг таркибида бир қатор функциялар аниқланганлиги бу тур билан ишлашда маълум бир қулайликлар яратади.

string туридаги ўзгарувчилар қуйидагича эълон қилиниши мумкин:

```
string s1,s2,s3;
```

Бу турдаги сатрлар учун махсус амаллар ва функциялар аниқланган.

string сатрга бошланғич қийматлар ҳар хил усуллар орқали бериш мумкин:

```
string s1="birinchi usul";
string s2("ikkinchi usul");
string s3(s2);
string s4=s2;
```

Худди шундай, string туридаги ўзгарувчилар устида қиймат бериш амаллари ҳам ҳар хил:

```
string s1,s2,s3; char *str="misol";
//satrli o'zgarmas qiymati berish
s1="Qiyamat berish 1-usul";
s2=str; // char turidagi satr yuklanmoqda
s3='A'; // bitta belgi qiymat sifatida berish
s3=s3+s1+s2+"0123abc"; //qiymat sifatida satr ifoda
```

Сатр элементида индекс воситасидан ташқари at() функцияси орқали мурожаат қилиш мумкин:

```
string s1="satr misoli";
cout<<s.at(3) // natijada 'r' belgisi ekranga chiqadi
```

Шуни айтиб ўтиш керакки, string синфда шу турдаги ўзгарувчилар билан ишлайдиган функциялар аниқланган. Бошқача айтганда, string турида эълон қилинган ўзгарувчилар (объектлар) ўз функцияларига эга ҳисобланади ва уларни чақириш учун олдин ўзгарувчи номи, кейин '.' (нуқта) ва зарур функция номи (аргументлари билан) ёзилади.

Қуйидаги жадвалда string туридаги сатрлар устида бажариладиган амаллар келтирилган.

## string туридаги сатрлар устидан амаллар

Амал	Мазмуни	Мисол
=, +=	Қиймат бериш амали	s="satr01234" s+="2satr000"
+	Сатрлар улаш амали (конкантенация)	s1+s2
==, !=, <, <=, >, >=	Сатрларни солиштириш амаллари	s1==s2    s1>s2 && s1!=s2
[]	Индекс бериш	s[4]
<<	Оқимга чиқариш	cout << s
>>	Оқимдан ўқиш	cin >> s (probelgacha)

### 20.7. Сатр қисмини бошқа сатрга нусхалаш функцияси

Бир сатр қисмини бошқа сатрга юклаш учун куйидаги функцияларни ишлатиш мумкин, уларни прототипи куйидагича:

```
assign(const string &str);
assign(const string &str,unsigned int pos,
        unsigned int n);
assign(const char *str, int n);
```

Биринчи функция қиймат бериш амал билан эквивалентдир: string туридаги стр сатр ўзгарувчи ёки сатр ўзгармасни амални чақирувчи сатрга беради:

```
string s1,s2;
s1="birinchi satr";
s2.assign(s1); // s2=s1 amalga ekvivalent
```

Иккинчи функция чақирувчи сатрга аргументдаги стр сатрнинг пос ўрнидан n та белгидан иборат бўлган сатр қисмини нусхалайди. Агарда пос қиймати стр сатр узунлигидан катта бўлса, хатолик ҳақида огоҳлантирилади, агар pos + n ифода қиймати стр сатр узунлигидан катта бўлса, стр сатрнинг пос ўрнидан бошлаб сатр охиригача бўлган белгилар нусхаланади. Бу қоида

барча функциялар учун тегишлидир.

Мисол:

```
string s1,s2,s3;
s1="0123456789";
s2.assign(s1,4,5);    // s2="45678"
s3.assign(s1,2,20);   // s3="23456789"
```

Учинчи функция аргументдаги чар туридаги стр сатрни string турига айлантириб, функцияни чакирувчи сатрга ўзлаштиради:

```
char * strold;
cin.getline(strold,100); //"0123456789" kiritiladi
string s1,s2;
s2.assign(strold,6);    // s2="012345"
s3.assign(strold,20);  // s3="0123456789"
```

Сатр қисмини бошқа сатрга қўшиш функциялари қуйидагича:

```
append(const string &str);
append(const string & str,unsigned int pos,
        unsigned int n);
append(const char *str, int n);
```

Бу функцияларни юқорида келтирилган мос ассигн функциялардан фарқи - функцияни чакирувчи сатр охирига стр сатрни ўзини ёки унинг қисмини қўшади.

```
char * sc;
cin.getline(sc,100);  //"0123456789" kiritiladi
string s1,s,s2;
s2=sc; s1="misol";
s="aaa";              //s2="0123456789"
s2.append("abcdef"); //s2+="abcdef" amali
                      //va s2="0123456789abcdef"
s1.append(s2,4,5);    //s1="misol45678"
s.append(ss,5);      // s="aaa012345"
```

Бир сатрга иккинчи сатр қисмини жойлаштириш учун қуйидаги функциялар ишлатилади:



```
insert(unsigned int pos1,const string &str);
insert(unsigned int pos1,const string & str,
        unsigned int pos2,unsigned int n);
insert(unsigned int pos1,const char *str, int n);
```

Бу функциялар append каби ишлайди, фарқи шундаки, стр сатрини ёки унинг қисмини функцияни чақирувчи сатрнинг кўрсатилган pos1 ўрнидан бошлаб жойлаштиради. Бунда амал чақирувчи сатрнинг pos1 ўриндан кейин жойлашган белгилар ўнга сурилади.

Мисол:

```
char * sc;
cin.getline (sc,100); //”0123456789” satri kiritiladi
unsigned int i=3;
string s1,s,s2;
s2=sc; s1=”misollar”; s=”xyz”; // s2=”0123456789”
s2.insert(i,”abcdef”); // s2=”012abcdef3456789”
s1.insert(i-1,s2,4,5); // s1=”mi45678sollar”
s.insert(i-2,sc,5); // s=”x01234yz”
```

## 20.8. Сатр қисмини ўчириш ва алмаштириш функциялари

Сатр қисмини ўчириш учун қуйидаги функцияни ишлатиш мумкин:

```
erase(unsigned int pos=0,unsigned int n=npos);
```

Бу функция, уни чақирувчи сатрнинг pos ўрнидан бошлаб n та белгини ўчиради. Агарда pos кўрсатилмаса, сатр бошидан бошлаб ўчирилади. Агар n кўрсатилмаса, сатрни охиригача бўлган белгилар ўчирилади:

```
string s1,s2,s3;
s1=”0123456789”;
s2=s1;s3=s1;
s1.erase(4,5); // s1=”01239”
s2.erase(3); // s2=”012”
s3.erase(); // s3=””
```

void clear() функцияси, уни чақирувчи сатрни тўлиқ тозалайди.

Масалан:

```
s1.clear(); //s1tr bo'sh hisoblanadi (s1="")
```

Бир сатр қисмининг ўрнига бошқа сатр қисмини қўйиш учун қуйидаги функциялардан фойдаланиш мумкин:

```
replace(unsigned int pos1,unsigned int n1,  
        const string & str);  
replace(unsigned int pos1,unsigned int n1,  
        const string & str,unsigned int pos2,  
        unsigned int n2);  
replace(unsigned int pos1,unsigned int n1,  
        const char *str, int n);
```

Бу функциялар insert каби ишлайди, ундан фарқли равишда амал чакирувчи сатрнинг кўрсатилган ўрндан (pos1) n1 белгилар ўрнига стр сатрини ёки унинг pos2 ўрндан бошланган n2 белгидан иборат қисмини қўяди (алмаштиради).

Мисол:

```
char * sc="0123456789";  
unsigned int i=3,j=2;  
string s1,s,s2;  
s2=sc; s1="misollar"; s="xyz"; // s2="0123456789"  
s2.replace(i,j,"abcdef"); // s2="012abcdef56789"  
s1.replace(i-1,j+1,s2,4,5); // s1="mi45678lar"  
s.replace(i-2,j+2,sc,5); // s="x012345"
```

swap (string & стр) функцияси иккита сатрларни ўзаро алмаштириш учун ишлатилади. Масалан:

```
string s1,s2;  
s1="01234";  
s2="98765432";  
s1.swap(s2); // s2="01234" va s1="98765432" bo'ladi.
```

Сатр қисмини ажратиб олиш функцияси

Функция прототиби қуйидагича:

```
string substr(unsigned int pos=0,
```

```
unsigned int n=npos)const;
```

Бу функция, уни чакирувчи сатрнинг пос ўрнидан бошлаб н белгини натижа сифатида қайтаради. Агарда пос кўрсатилмаса, сатр бошидан бошлаб ажратиб олинади, агар н кўрсатилмаса, сатр охиригача бўлган белгилар натижа сифатида қайтарилади:

```
string s1,s2,s3;  
s1="0123456789";  
s2=s1; s3=s1;  
s2=s1.substr(4,5); // s2="45678"  
s3=s1.substr(3); // s3="3456789"  
// "30123456789" satr ekranga chiqadi  
cout<<s1.substr(1,3)+s1.substr();
```

## 20.9. Сатр қисмини излаш ва солиштириш функциялари

string синфида сатр қисмини излаш учун ҳар хил вариантдаги функциялар аниқланган. Қуйида улардан асосийларининг тавсифини келтирамиз.

```
unsigned int find(const string &str,  
                unsigned int pos=0)const;
```

Функция, уни чақирган сатрнинг кўрсатилган жойдан (pos) бошлаб стр сатрни қидиради ва биринчи мос келувчи сатр қисмининг бошланиш индексини жавоб сифатида қайтаради, акс ҳолда максимал мусбат бутун нпос сонни қайтаради (npos=4294967295), агар излаш ўрни (pos) берилмаса, сатр бошидан бошлаб изланади.

```
unsigned int find(char c, unsigned int pos=0)const;
```

Бу функция олдингидан фарқи равишда сатрдан с белгисини излайди.

```
unsigned int rfind(const string &str,  
                unsigned int pos=npos)const;
```

Функция, уни чақирган сатрнинг кўрсатилган пос ўрнигача стр сатрнинг биринчи учраган жойини индексини қайтаради, акс ҳолда нпос қийматини қайтаради, агар пос кўрсатилмаса сатр охиригача излайди.

```
unsigned int rfind(char c, unsigned int pos=npos) const;
```

Бу функциянинг олдингидан фарқи - сатрдан с белгиси изланади.

```
unsigned int find_first_of(const string &str,  
                           unsigned int pos=0)const;
```

Функция, уни чақирган сатрнинг кўрсатилган (пос) жойдан бошлаб стр сатрининг ихтиёрий бирорта белгисини қидиради ва биринчи учраганининг индексини, акс ҳолда нпос сонини қайтаради.

```
unsigned int find_first_of(char c,  
                           unsigned int pos=0)const;
```

Бу функциянинг олдингидан фарқи - сатрдан с белгисини излайди;

```
unsigned int find_last_of(const string &str,  
                          unsigned int pos=npos)const;
```

Функция, уни чақирган сатрнинг кўрсатилган (пос) жойдан бошлаб стр сатрни ихтиёрий бирорта белгисини қидиради ва ўнг томондан биринчи учраганининг индексини, акс ҳолда нпос сонини қайтаради.

```
unsigned int find_last_of(char c,  
                          unsigned int pos=npos) const;
```

Бу функция олдингидан фарқи - сатрдан с белгисини излайди;

```
unsigned int find_first_not_of(const string &str,  
                               unsigned int pos=0)const;
```

Функция, уни чақирган сатрнинг кўрсатилган (пос) жойдан бошлаб стр сатрнинг бирорта ҳам белгиси кирмайдиган сатр қисмини қидиради ва чап томондан биринчи учраганининг индексини, акс ҳолда нпос сонини қайтарилади.

```
unsigned int find_first_not_of(char c,  
                               unsigned int pos=0)const;
```

Бу функциянинг олдингидан фарқи - сатрдан с белгисидан фарқли биринчи белгини излайди;

```
unsigned int find_last_not_of(const string &str,  
                              unsigned int pos=npos)const;
```

Функция, уни чақирувчи сатрнинг кўрсатилган жойдан бошлаб стр сатрини ташкил этувчи белгилар тўпламига кирмаган белгини қидиради ва энг ўнг томондан биринчи топилган белгининг индексини, акс ҳолда нпос сонини қайтаради.

```
unsigned int find_last_not_of(char c,
```

```
unsigned int pos=npos)const;
```

Бу функциянинг олдингидан фарқи - сатр охиридан бошлаб с белгисига ўхшамаган белгини излайди.

Излаш функцияларини қўллашга мисол:

```
#include <iostream>
using namespace std;
#include <conio.h>
void main()
{
    string s1="01234567893456ab2csef",
           s2="456",s3="ghk2";
    int i,j;
    i=s1.find(s2);
    j=s1.rfind(s2);
    cout<<i; // i=4
    cout<<j; // j=11
    cout<<s1.find('3') <<endl; // natija 3
    cout<<s1.rfind('3') <<endl; // natija 10
    cout<<s1.find_first_of(s3)<<endl; // natija 2
    cout<<s1.find_last_of(s3)<<endl; // natija 16
    cout<<s1.find_first_not_of(s2)<<endl; // natija 14
    cout<<s1.find_last_not_of(s2)<<endl; // natija 20
}
```

Сатрлар қисмларини солиштириш учун compare() функцияси ишлатилади:

```
int compare(const string &str)const;
int compare(unsigned int pos1,unsigned int n1,
            const string & str)const;
int compare(unsigned int pos1,unsigned int n1,
            const string & str,unsigned int pos2,
            unsigned int n2)const;
```

Функциянинг биринчи шаклида иккита сатрлар тўла солиштирилади:

функция манфий сон қайтаради, агар функцияни чақирувчи сатр стр сатрдан кичик бўлса, 0 қайтаради агар улар тенг бўлса ва мусбат сон қайтаради, агар функция чақирувчи сатр стр сатрдан катта бўлса.

Иккинчи шаклда худди биринчидек амаллар бажарилади, фақат функция чақирувчи сатрнинг pos1 ўрнидан бошлаб n1 та белгили сатр ости стр сатр билан солиштирилади.

Учинчи кўринишда функция чақирувчи сатрнинг pos1 ўрнидан бошлаб n1 та белгили сатр қисми ва стр сатрдан pos2 ўрнидан бошлаб n2 та белгили сатр қисмлари ўзаро солиштирилади.

Мисол:

```
#include <iostream>
using namespace std;
void main()
{
    String s1="01234567893456ab2csef", s2="456",
           s3="ghk";
    cout<<"s1="<<s1<<endl;
    cout<<"s2="<<s2<<endl;
    cout<<"s3="<<s3<<endl;
    if(s2.compare(s3)>0)cout<<"s2>s3"<<endl;
    if(s2.compare(s3)==0)cout<<"s2=s3"<<endl;
    if(s2.compare(s3)<0)cout<<"s2<s3"<<endl;
    if(s1.compare(4,6,s2)>0)cout<<"s1[4-9]>s2"<<endl;
    if(s1.compare(5,2,s2,1,2)==0)
        cout<<"s1[5-6]=s2[1-2]"<<endl;
}
```

Сатр хоссаларини аниқлаш функциялари

string синфида сатр узунлиги, унинг бўшлигини ёки эгаллаган хотира хажмини аниқлайдиган функциялар бор:

```
unsigned int size()const; // satr o'lchami
unsigned int length()const; // satr elementlar soni
unsigned int max_size()const; // satrning maksimal
```

```

// uzunligi(4294967295)
unsigned int capacity()const;// satr egallagan хотира
// hajmi
bool empty()const; // true, agar satr bo'sh bo'lsa
string туридаги сатрни чар турига ўтказиш учун
const char * c_str()const;

```

функцияни ишлатиш керак. Бу функция чар турдаги `\0` белгиси билан тугайдиган сатрга ўзгармас кўрсаткични қайтаради:

```

char *s1; string s2="0123456789";
s1=s2.c_str();

```

Худди шу мақсадда

```

const char * data()const;

```

функциясидан ҳам фойдаланиш мумкин. Лекин бу функция сатр охирига `\0` белгисини қўшмайди.

#### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

#### Мустақил таълим мавзулари:

1. Сатрларни ҳосил қилиш усуллари.
2. Сатрлар массивини ишлатиш.
3. Сатрларни солиштириш функциялари.
4. Сатрлар билан ишловчи киритиш-чиқариш функциялари.

## Глоссарий

Термин	Terminology	Ўзбек тилидаги шархи
include	include	препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
Функция	Function	программа алоҳида бўлаги, асосий қисм томонидан чақириб ишлатилади
Параметр	Parametr	функция ишлаши учун керак берилганлар
аргумент	Argument	функцияга параметрига жўнатиладиган қиймат
Динамик	Dinamic	қиймати дастур ишўлаши давомида ўзгариб турадиган берилган
new	New	хотирадан янги жой ажратиш
delete	Delete	хотирадан ажратилган жойни тозалаш
Кўрсаткич	pointer	қийматлари адрес бўлган ўзгарувчилар
cout	cout	экранга чиқариш оқими.
cin	cin	экрандан киритиш оқими.
Ўзгарувчи	Variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги.
Идентификатор	identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизиқ ('_') белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги.

## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур киритилган сатрдаги сўзларни алоҳида қатордан чоп қилиши керак. Лекин дастур нотўғри натижа бермоқда. Сўз сифатида бўш жойлар ҳам чоп этилмоқда. Дастурни тўғри ишлашини таъминлаш керак.



### **Кейсни бажариш босқчилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

### **Назорат саволлари:**

1. Массив тушунчасига таъриф беринг.
2. Қандай турдаги массивларни биласиз?
3. Массивлардан қандай мақсадларда фойдаланиш мумкин?
4. Сонлар массиви функция параметри сифатида қандай жўнатилади?
5. Бир ўлчамли массив функция параметри сифатида ишлатишга мисол келтиринг.
6. Кўп ўлчамли массивларни параметр сифатида ишлатишда қандай усуллардан фойдаланиш мумкин?
7. Берилган қийматни массив элементларидан қидириш функциясини қуринг.
8. Ўзгарувчан параметрли функциялар сарлавхаси қандай ёзилади?
9. Ўзгарувчан сондаги параметрларни ташкил қилишнинг қандай усуллари мавжуд ва улар қай тарзда амалга оширилади?
10. Матрица киритиш ва чиқариш функцияларини қуринг.

### **Тест саволлари:**

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="ABBA", s2[]="BC", s3[]="CCBA";  
strncat(s2,s3,2);  
strcpy(s1,s2);  
cout << s2;
```

- a) ABC
- b) BCCC
- c) CBVV
- d) BCCA

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="BAB", s2[]="ABC", s3[]="CBA";  
string a="ABC";  
if (s1[0]==s1[1] && s2[0])  
cout<<s3;  
else cout << a[1];
```

- a) B
- b) C
- c) A
- d) AB

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
string s1="AB", s2="BCC", s3="CAC";  
int i=1;  
s1.insert(i,s3);  
s2.append(s3);  
s2.erase(1,2);  
s3.erase(2,1);  
s3=s1+s2.substr(1,1);  
cout<<s3;
```

- a) ACACBC
- b) ABACAB
- c) BCBAVC
- d) ABCABC

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="BAB", s2[]="ABC", s3[]="CBA";  
if (s1[0]==s1[1] && s3[0])  
cout<<s3;  
else cout << s2;
```

- a) ABC
- b) CBA
- c) BAC
- d) ACB

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char c[]={'a','i','o','e','\0'};  
char matn[] = "Programmash asoslari va kompyuter texnologiyalari";  
int k=0;  
for(int i=0; i<strlen(c);i++){  
    k=0;  
    for(int j=0;j<strlen(matn);j++){  
        if(matn[j]==c[i])  
            k++;  
    }  
    cout<<k<<"\t";  
}
```

- a) 8 3 5 2
- b) 2 3 5 8
- c) 0 3 5 8
- d) a i o e

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="BAB", s2[]="ABC", s3[]="ABA";  
string a="lalalal";  
if (strrev(s3)==s3)  
cout<<strrev(strcpy(s2,strrev(s1)));  
else cout << a.find("al");
```

- a) BAB
- b) ABA
- c) ABC
- d) CBA

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int s=0, m=0 ;
char matn[] = "Programmash asoslari fani 2015 - yil";
    for(int i=0;i<strlen(matn);i++)
        if(isdigit(matn[i])) s++;
        else if (isalpha(matn[i])) m++;
    cout<<s<<"\t"<<m;
```

- a) 4 28
- b) 2 31
- c) 5 29
- d) 6 23

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="ABBA", s2[]="BABC", s3[]="CCBA";
strncpy(s2,s3,2);
strncpy(s1,s2,1);
cout << s1;
```

- a) CBBA
- b) ABBC
- c) ACCB
- d) BAAC

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
string s1="A", s2="B", s3="C";
int i=1;
s1.insert(i,s3);
s1.append(s2);
```

```
s2.insert(i,s3);  
s1.erase(1,2);  
s3=s1+s2;  
cout<<s3;
```

- a) ABC
- b) CAB
- c) ACB
- d) BBC

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char matn[] = "d2as#3!";  
for(int i=0;i<strlen(matn)-2;i++)  
if(!isdigit(matn[i]) && isalpha(matn[i]))  
cout<<matn[i];
```

- a) das
- b) asd
- c) 2#3
- d) 23

## V. АМАЛИЙ МАШҒУЛОТ МАТЕРИАЛЛАРИ

### Амалий машғулот 1.

#### C++ тили синтаксиси ва унинг лексик асоси

**Ишдан мақсад:** C++ дастурлаш тили дастур тузиш кўникмалари билан танишиш. Содда масалалар алгоритмини блок-схема кўринишида тасвирлаш, Visual Studio 2013 дастурлаш мухитида C++ тили орқали масаланинг дастурини тузиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани алгоритмини блок-схема кўринишида тасвирлаши, C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

#### Ишни бажариш учун намуна

**Мисол:** Берилган  $x$ ,  $y$  ва  $z$  сонлари учун формуланинг натижаси топилсин.

Берилганлар:  $x=14.26$ ,  $y=-1.22$ ,  $z=3.5 \times 10^{-2}$

$$t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{0.5 + \sin^2 y} \left(1 + \frac{z^2}{3 - z^2 / 5}\right)$$

Натижа:  $t=0.564849$

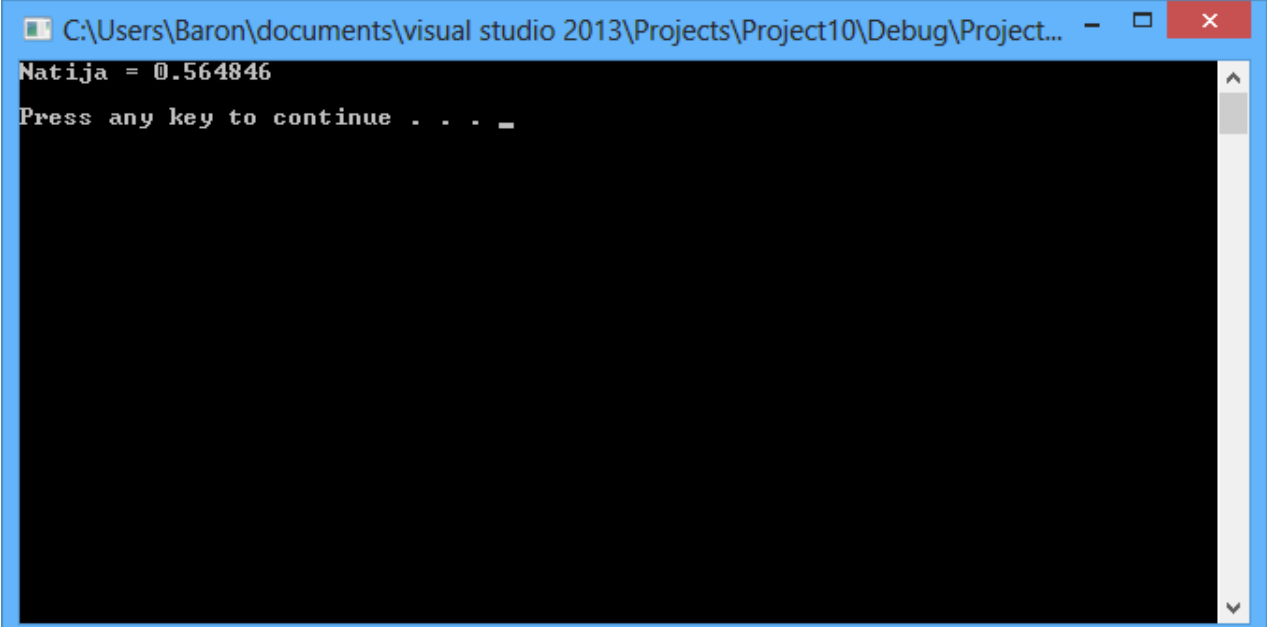
Дастур коди:

dastur.cpp файли:

```
#define _USE_MATH_DEFINES
#include<iostream>
#include<cmath>
using namespace std;
int main(){
    double x,y,z,a,b,c,t;
    x=14.26;
    y=-1.22;
```

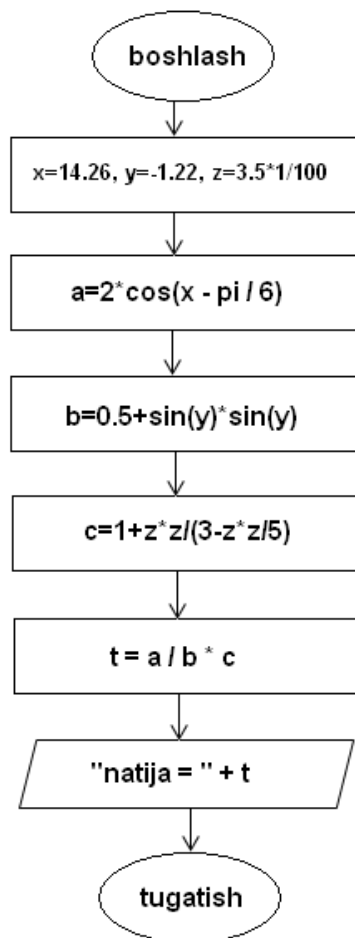
```
z=3.5*1/100.;  
a=2*cos(x-M_PI/6);  
b=0.5+sin(y)*sin(y);  
c=1+z*z/(3-z*z/5);  
t=a/b*c;  
cout << "Natija = " << t << endl;  
system("pause");  
return 0;  
}
```

Дастур ишлаши натижаси:



```
C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...  
Natija = 0.564846  
Press any key to continue . . . _
```

Масала алгоритми блок схемаси:



### Амалий топшириқлар

Берилган  $x$ ,  $y$  ва  $z$  сонлари учун формуланинг натижаси топилсин.

1.  $x=14.26, y=-1.22, z=3.5 \times 10^{-2},$

$$t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{0.5 + \sin^2 y} \left(1 + \frac{z^2}{3 - z^2 / 5}\right)$$

Натижа:  $t=0.564849$

2.  $x=-4.5, y=0.75 \times 10^{-4}, z=0.845 \times 10^2,$

$$u = \frac{\sqrt[3]{8 + |x - y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x-y|} (tg^2 z + 1)^x.$$

Натижа:  $u=-55.6848.$

3.  $x=-15.246, y=4.642 \times 10^{-2}, z=20.001 \times 10^2,$

$$\alpha = \ln\left(y^{-\sqrt{|x|}}\right) \left(x - \frac{y}{2}\right) + \sin^2 \arctg(z).$$

Натижа:  $\alpha=-182.036$



4.  $x=0.1722, y=6.33, z=3.25 \times 10^{-4},$

$$\gamma = 5 \operatorname{arctg} x - \frac{1}{4} \operatorname{arccos} x \frac{x + 3|x - y| + x^2}{|x - y|z + x^2}.$$

Натижа:  $\gamma = -172.025$

5.  $x=1.825 \times 10^2, y=18.225, z=-3.298 \times 10^{-2},$

$$\psi = \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right| + (y - x) \frac{\cos y - \frac{z}{(y - x)}}{1 + (y - x)^2}.$$

Натижа:  $\psi = 1.2131$

6.  $x=6.251, y=0.827, z=25.001,$

$$b = y^{\sqrt[3]{|x|}} + \cos^3 y \frac{|x - y| \left( 1 + \frac{\sin^2 z}{\sqrt{x + y}} \right)}{e^{|x - y|} + x/2}.$$

Натижа:  $b = 0.7121$

7.  $x=17.421, y=10.365 \times 10^{-3}, z=0.828 \times 10^5,$

$$f = \frac{\sqrt[4]{y + \sqrt[3]{x - 1}}}{|x - y|(\sin^2 z + \operatorname{tg} z)}.$$

Натижа:  $f = 0.33056$

8.  $x=2.444, y=0.869 \times 10^{-2}, z=-0.13 \times 10^3,$

$$h = \frac{x^{y+1} + e^{y-1}}{1 + x|y - \operatorname{tg} z|} \left( 1 + |y - x| \right) + \frac{|y - x|^2}{2} - \frac{|y - x|^3}{3}.$$

Натижа:  $h = -0.49871$

9.  $x=1, y=1, z=3$

$$a = (1 + y) \frac{x + y/(x^2 + 4)}{e^{-x-2} + 1/(x^2 + 4)};$$

$$b = \frac{1 + \cos(y - 2)}{x^4 / 2 + \sin^2 z}.$$

Натижа:  $a = 9.608184; b = 2.962605$

10.  $x=3, y=4, z=5,$

$$a = \frac{1 + \sin^2(x + y)}{2 + \left| x + \frac{2x}{(1 + x^2 y^2)} \right|} + x;$$

$$b = \cos^2\left(\operatorname{arctg} \frac{1}{z}\right).$$

Натижа: a=3.288716; b=0.9615385

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Тарқатма материаллар

#### 1. Калит сўзлар

asm, auto, break, case, catch, char, class, const, continue, default, delete, do, double, else, enum, explicit, extern, float, for, friend, goto, if, inline, int, long, mutable, new, operator, private, protected, public, register, return, short, signed, sizeof, static, struct, switch, template, this, throw, try, typedef, typename, union, unsigned, virtual, void, volatile, while.

#### 2. Процессор регистрлари

\_AH, \_AL, \_AX, \_EAX, \_BH, \_BL, \_BX, \_EBX, \_CL, \_CH, \_CX, \_ECX, \_DH, \_DL, \_DX, \_EDX, \_CS, \_ESP, \_EBP, \_FS, \_GS, \_DI, \_EDI, \_SI, \_ESI, \_BP, \_SP, \_DS, \_ES, \_SS, \_FLAGS.

## Кейслар банки

**Кейс 1.** Берилган формулага асосан ишлайдиган дастур тузилди. Лекин дастурнинг компиляция жараёнида хатолик борлиги аниқланди.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 7, b = 8;
```

```
cout << a % b;
```

- a) 7
- b) 8
- c) 1
- d) 56

2. С++ тилида экранга чиқариш оқими ишлатилиши тўғри кўрсатилган жавобни танланг.

- a) `cout << a;`
- b) `cout >> a;`
- c) `cout <<<< a;`
- d) `cin >> a;`

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int s=2468;
```

```
cout<<(s%1000)/10;
```

- a) 46
- b) 68
- c) 468
- d) 24

4. Идентификатор сифатида фойдаланиш мумкин бўлмаган жавобни кўрсатинг?

- a) delete
- b) dlete
- c) intt
- d) filloat

5. Идентификатор сифатида фойдаланиш мумкин бўлган жавобни кўрсатинг а)  
inT

- b) float
- c) for
- d) double

6. Идентификатор сифатида фойдаланиш мумкин бўлган жавобни кўрсатинг?

- a) const
- b) include
- c) abc
- d) abs

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=6789;  
cout<<a/100;
```

- a) 67
- b) 89
- c) 67.89
- d) 89.67

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
float a=1234;  
cout<<a/100;
```

- a) 12.34
- b) 12
- c) 34
- d) 34.12

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
float a=1234;
```

```
cout<< a+1%100;
```

a) 35

b) 1235

c) 12

d) 100

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=456;
```

```
cout<<a+a/100;
```

a) 460

b) 456

c) 460.56

d) 4.56

## Амалий машғулот 2.

### C++ тили дастурининг тузилиши ва шакли

**Ишдан мақсад:** C++ дастурлаш тилида дастурнинг тузилишини билан ишлаш, тўғри шаклда дастур кодини шакллантириш, форматлаш, номланган идентификаторлардан фойдалана олиш.

**Масаланинг қўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Берилган сон секунд сифатида қаралсин ва секунд соат, минут ва секунд кўринишида ифодалансин.

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;

int main(){
    int sekund, minut, soat;
    cout << "Sekund = ";
    cin >> sekund;
    soat = sekund / 3600;
    sekund = sekund - soat * 3600;
    minut = sekund / 60;
    sekund = sekund - minut * 60;
    cout << "Soat = " << soat << endl;
    cout << "Minut = " << minut << endl;
    cout << "Sekund = " << sekund << endl;
    system("pause");
    return 0;
}
```

Дастур ишлаши натижаси:

```
C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
Sekund = 5698
Soat = 1
Minut = 34
Sekund = 58
Press any key to continue . . . _
```

### Амалий топшириқлар

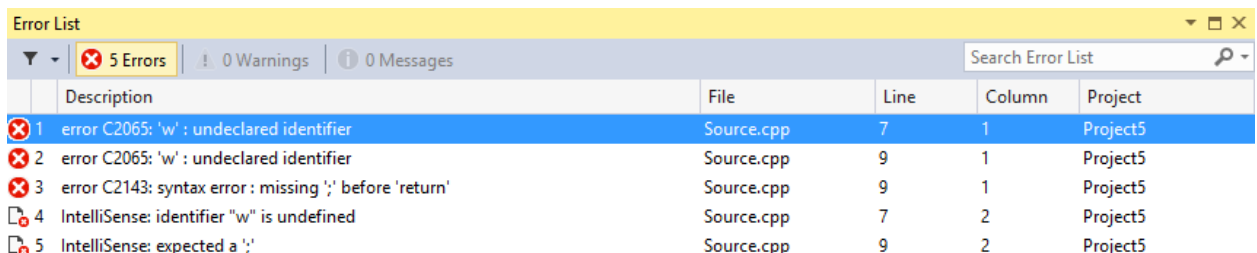
1. Берилган  $R$  радиусли айлананинг узунлиги, доиранинг юзаси ва шарнинг ҳажми ҳисоблансин.
2. Учбурчак учларининг координаталари асосида унинг юзаси ва периметри топилсин.
3. Берилган тўрт хонали сон рақамларининг кўпайтмаси топилсин.
4. Массалари мос равишда  $m_1$  ва  $m_2$  бўлган ва бир биридан  $r$  масофада жойлашган иккита жисмнинг ўзаро тортишиш кучи  $F$  аниқлансин.
5. Тўғрибурчакли учбурчакнинг гипотенузаси ва катети берилган бўлса, иккинчи катет ва ички чизилган айлананинг радиуси ҳисоблансин.
6. Айлана узунлиги берилган, бу айлана билан чегараланган доира юзаси топилсин.
7. Учбурчак бурчакларининг катталиклари ва ички чизилган айлана радиуси билан берилган. Учбурчак томонлари узунликлари ҳисоблансин.
8. Учбурчак томонлари берилган. Қуйидагилар аниқлансин:
  - а) баландликлари;      б) медианалари;
  - е) биссектрисалари;    д) ички ва ташқи чизилган айлана радиуслари.
9. Соат милининг сутка бошига нисбатан  $f$  градус бурилишига мос келувчи  $h$  соат,  $m$  минут ва  $s$  секунд топилсин ( $0 \leq f \leq 360^\circ$ ,  $f$  - ҳақиқий сон).

## Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

### 1.



	Description	File	Line	Column	Project
1	error C2065: 'w' : undeclared identifier	Source.cpp	7	1	Project5
2	error C2065: 'w' : undeclared identifier	Source.cpp	9	1	Project5
3	error C2143: syntax error : missing ';' before 'return'	Source.cpp	9	1	Project5
4	IntelliSense: identifier "w" is undefined	Source.cpp	7	2	Project5
5	IntelliSense: expected a ';'	Source.cpp	9	2	Project5

### Дастур хатоларини кўриш ойнаси

2. Хужжатлашган идентификаторлардан фойдаланиш.

```
const double SANTIMETR_UCHUN_MILLIMETR = 0.1;
double santimetr;
double millimetr;
santimetr = millimetr * SANTIMETR_UCHUN_MILLIMETR;
```

## Кейслар банки

**Кейс 1.** Дастур тузилди ва унинг вазифаси киритилган граммга (<1000000) нисбатан унинг тоннадаги кўринишини аниқлашдан иборат. Дастур ишлаши натижасида факт нол жавоб чоп этмоқда.

### Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).



- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тилида тўғри ишлайдиган дастур кодини тақдим этинг (индивидуал ҳолда).

### Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int c=25/3
cout << c;
```

- a) 8
- b) 25/3
- c) 8.3
- d) 8.(3)

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int c=14/4
cout << 8+c;
```

- a) 11
- b) 11.5
- c) 5.5
- d) 8+c

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int d=3
cout << d/2;
```

- a) 1
- b) 1.5
- c) d
- d) 3/2

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int d=3  
cout << d/2.;
```

- a) 1
- b) 1.5
- c) d
- d) 3/2

5. Қуйидаги программа қисмидаги хато кодни топинг.

```
const int n=3;  
int a=n;  
a=a+2;  
n=n+2;
```

- a) n=n+2;
- b) a=a+2;
- c) int a=n;
- d) const int n=3;

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int s=0;  
int a=s+2;  
cout<<s;
```

- a) 0
- b) 2
- c) s
- d) a

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a, b, c;  
a=b=c=2;  
c=c+b;  
cout << c;
```

- a) 4
- b) 2
- c) 6
- d) c

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int x, y, z=2;  
x=z;  
y=z=3;  
cout << x+z;
```

- a) 5
- b) 4
- c) x+z
- d) 6

9. Қуйидаги программа қисмидаги хато кодни топинг?

```
int a=2,b;  
b=a+3;  
cout << "a++b";  
cout << a++b;
```

- a) cout << a++b;
- b) cout << "a++b";
- c) b=a+3;
- d) int a=2,b;

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=2,b;  
b=a+3;  
cout << a+b;
```

- a) 7
- b) 5
- c) 2
- d) 3

### Амалий машғулот 3.

#### Берилганлар турлари. C++ тилининг таянч турлари

**Ишдан мақсад:** C++ дастурлаш тилида берилганларнинг турли кўринишларидан фойдаланишни ўрганиш, улардан фойдалана олиш.

**Масаланинг қўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

#### Ишни бажариш учун намуна

**Мисол:** Учта томони билан берилган учбурчакнинг периметри ва юзаси топилсин.

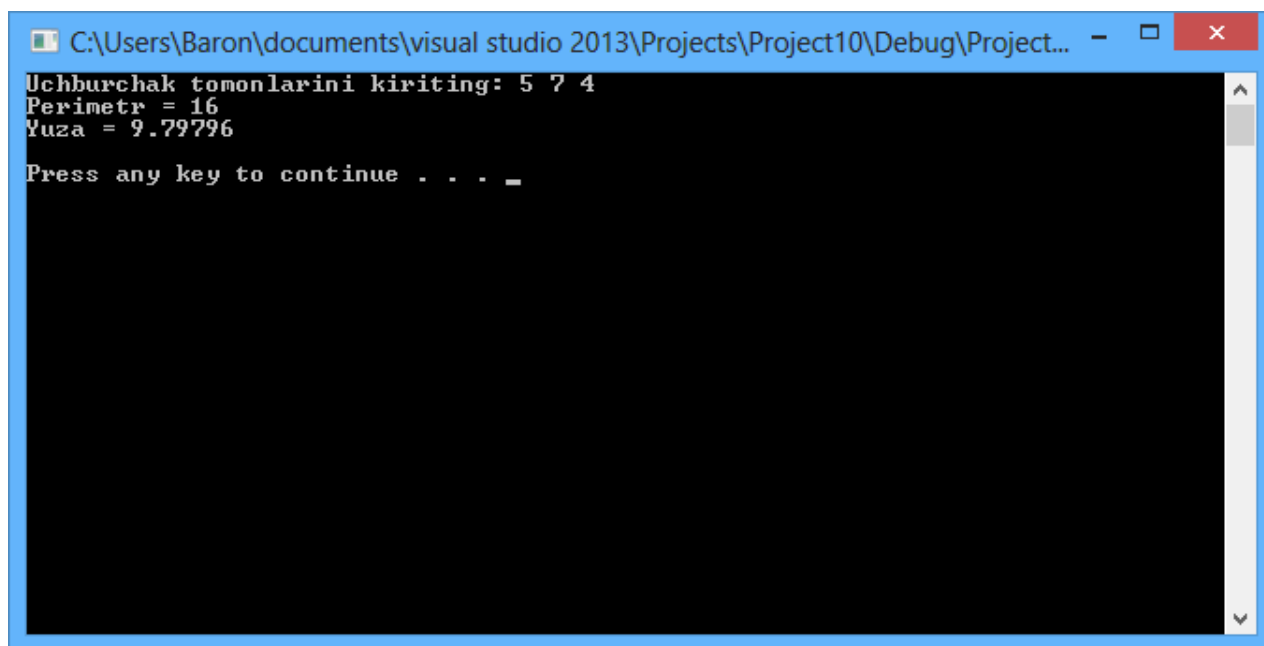
Дастур коди:

dastur.cpp файли:

```
#include <iostream>
#include <cmath>
using namespace std;

int main(){
    int a, b, c;
    double p, s;
    cout << "Uchburchak tomonlarini kiriting: ";
    cin >> a >> b >> c;
    p = (a + b + c) / 2.;
    s = sqrt(p * (p-a) * (p-b) * (p-c));
    cout << "Perimetr = " << p*2 << endl;
    cout << "Yuza = " << s << endl;
    system("pause");
    return 0;
}
```

Дастур ишлаши натижаси:



```
C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...  
Uchburchak tomonlarini kiriting: 5 7 4  
Perimetr = 16  
Yuza = 9.79796  
Press any key to continue . . . _
```

### Амалий топшириқлар

1. Турғун сувдаги қайиқ тезлиги  $V$  км/с. Дарё суви оқимининг тезлиги  $U$  км/с ( $U < V$ ). Қайиқ кўлда  $T_1$  соат, дарёда эса (оқимга қарши)  $T_2$  соат ҳаракат қилган. Қайиқ сузган умумий  $S$  масофа топилсин.
2. Биринчи автомобил тезлиги  $V_1$  км/с, иккинчисиники -  $V_2$  км/с, улар орасидаги масофа -  $S$  км. Автомобиллар бир-биридан узоқлашса (бир-бирига қараб ҳаракат қилганда),  $T$  соатдан кейин улар орасидаги масофа қандай бўлади?
3. Асослари  $a$  ва  $b$  ( $a > b$ ), катта асосдаги бурчаги  $\alpha$  бўлган тенг ёнли трапетсиянинг периметри ҳамда юзаси топилсин (бурчак радианда берилди).
4. Нолдан фарқли берилган  $R_1, R_2, R_3$  электр қаршиликлари учун  $R$  ҳисоблансин. Бунда: 
$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}.$$
5. Ходимнинг ойлик иш ҳақиға 45% мукофот пули қўшилсин. Ҳосил бўлган миқдордан 17% даромад солиғи, 1,5% касаба уюшмаси ва 1% нафақа солиғи ушлаб қолинсин. Қўлга тегадиган пул миқдори чоп этилсин.
6. Уч хонали бутун сон ( $k$ ) рақамлари йиғиндисини ( $s$ ) бутун ўзгарувчига ўзлаштирилсин.
7. Тенг томонли учбурчак томони берилган, учбурчак юзаси топилсин.

8. Учта мусбат сон берилган. Сонлар ўрта геометригининг каср қисми топилсин.
9. Берилган катетлари бўйича тўғри бурчакли учбурчакнинг периметри ва юзаси ҳисоблансин.
10. Берилган икки томони ва улар орасидаги бурчак (градусда) асосида учбурчакнинг учинчи томони ва юзаси топилсин.
11. Берилган уч хонали сон рақамларини тескари тартибда ёзишдан ҳосил бўлган сон топилсин. Масалан, 345 сонининг тескари тартиби 543 бўлади.

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Тарқатма материаллар.

#### 1. Берилганлар турлари жадвали

Тур номи	Байтлардаги ўлчами	Қиймат чегараси
bool	1	true yoki false
unsigned short int	2	0..65535
short int	2	-32768..32767
unsigned long int	4	0..42949667295
long int	4	-2147483648..2147483647
int (16 razryadli)	2	-32768..32767
int (32 razryadli)	4	-2147483648..2147483647
unsigned int (16 razryadli)	2	0..65535
unsigned int (32 razryadli)	4	0..42949667295
unsigned char	1	0..255
char	1	-128..127

## 2. C++ тилида escape-белгилар жадвали.

escape-белгилар	Ички код (16 лик сон)	Номи	Амал
\\	0x5C	\	Тескари ён чизиқни чоп этиш
\'	0x27	'	Апострофни чоп этиш
\"	0x22	"	Қўштирноқни чоп этиш
\?	0x3F	?	Сўроқ белгиси
\a	0x07	bel	Товуш сигналини бериш
\b	0x08	bs	Курсорни 1 белги орқага қайтариш
\f	0x0C	ff	Саҳифани ўтказиш
\n	0x0A	lf	Қаторни ўтказиш
\r	0x0D	cr	Курсорни айна қаторнинг бошига қайтариш
\t	0x09	ht	Навбатдаги табуляция жойига ўтиш
\v	0x0D	vt	Вертикал табуляция
\000	000		Белги саккизлик коди билан берилганда
\xNN	0xNN		Белги ўн олтилик коди билан берилганда

### Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур бутун сонлар бўлинмасининг қолдиқ қисмини топади. Энди ушбу масалани ҳақиқий сон учун ишлайдиган қилиш керак. Соннинг тури ҳақиқий сон турига ўтказилса дастур компиляция хатолигини юзага келтирмоқда.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int c=25/3  
cout << c;
```

- a) 8
- b) 25/3
- c) 8.3
- d) 8.(3)

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int c=14/4  
cout << 8+c;
```

- a) 11
- b) 11.5
- c) 5.5
- d) 8+c

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int d=3  
cout << d/2;
```

- a) 1
- b) 1.5
- c) d
- d) 3/2

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int d=3  
cout << d/2.;
```

- a) 1
- b) 1.5
- c) d
- d) 3/2



5. Қуйидаги программа қисмидаги хато кодни топинг.

```
const int n=3;
```

```
int a=n;
```

```
a=a+2;
```

```
n=n+2;
```

a) n=n+2;

b) a=a+2;

c) int a=n;

d) const int n=3;

6. Идентификатор сифатида фойдаланиш мумкин бўлган жавобни кўрсатинг?

a) const

b) include

c) abc

d) abs

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=6789;
```

```
cout<<a/100;
```

a) 67

b) 89

c) 67.89

d) 89.67

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
float a=1234;
```

```
cout<<a/100;
```

a) 12.34

b) 12

c) 34

d) 34.12

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
float a=1234;
```

```
cout<< a+1%100;
```

a) 35

b) 1235

c) 12

d) 100

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=456;
```

```
cout<<a+a/100;
```

a) 460

b) 456

c) 460.56

d) 4.56

## Амалий машғулот 4. Ўзгарувчилар ва ифодалар

**Ишдан мақсад:** C++ тили дастурларида ўзгарувчиларни ишлатиш, ифодаларни ёзишни ўрганиш. Мураккаб ифодаларни дастурини тузиш ва ҳисоблаш.

**Масаланинг қўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:**

$$a = \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right| + (y - x) \frac{\cos y - \frac{z}{(y-x)}}{1 + (y-x)^2}$$

ифода ҳисоблансин

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;
#include<math.h>

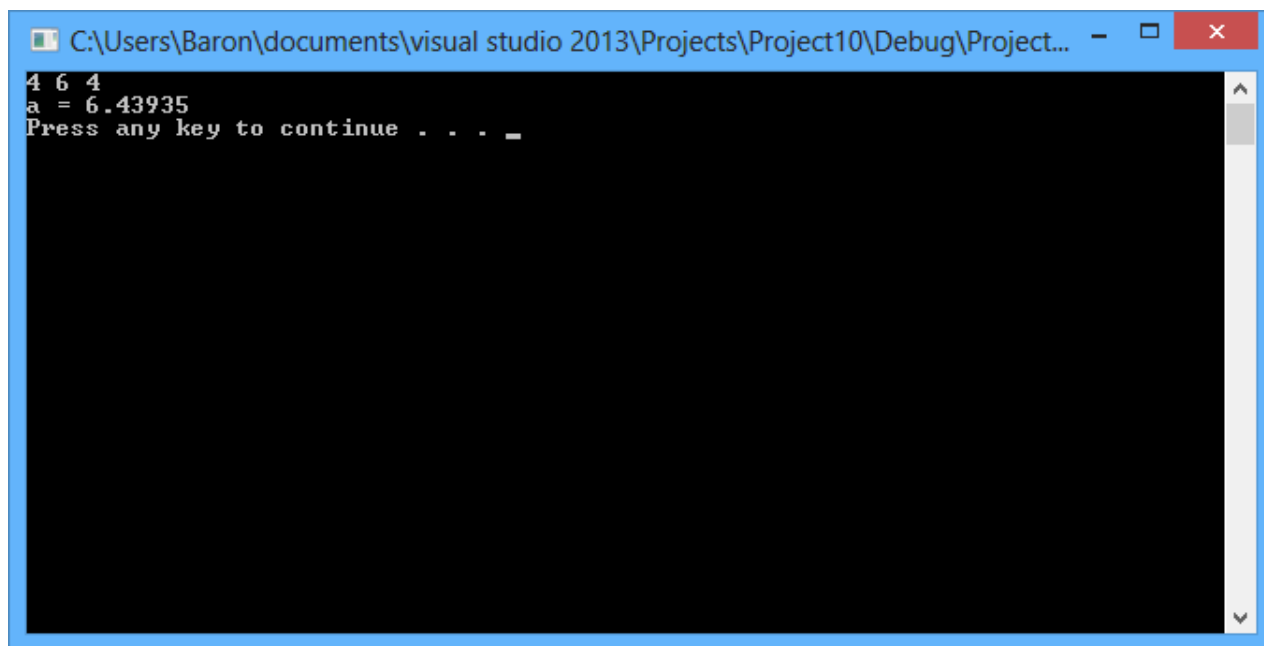
int main()
{
    float x, y, z, a, a1, a2, k, t;
    cin >> x >> y >> z;
    k = y / x;
    t = y - x;
    a1 = fabs(pow(x, k) - pow(k, 1. / 3));
    a2 = (cos(y) - z / t) / (1 + pow(t, 2));
    a = a1 + t*a2;
    cout << "a = " << a;
```

```

system("pause");
return 0;
}

```

Дастур ишлаши натижаси:



### Амалий топшириқлар

1. Берилганлар:  $x=3.74 \times 10^{-2}$ ,  $y=-0.825$ ,  $z=0.16 \times 10^2$ ,

$$v = \frac{1 + \sin^2(x + y)}{\left| x - \frac{2y}{1 + x^2 y^2} \right|} x^{|y|} + \cos^2\left(\operatorname{arctg} \frac{1}{z}\right)$$

Натижа:  $v=1.0553$

2. Берилганлар:  $x=0.4 \times 10^4$ ,  $y=-0.875$ ,  $z=-0.475 \times 10^{-3}$ ,

$$w = |\cos x - \cos y|^{(1+2\sin^2 y)} \left( 1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4} \right).$$

Натижа:  $w=1.9873$

3. Берилганлар:  $x=16.55 \times 10^{-3}$ ,  $y=-2.75$ ,  $z=0.15$ ,

$$\beta = \sqrt{10(\sqrt[3]{x} + x^{y+2})} (\arcsin^2 z - |x - y|).$$

Натижа:  $\beta = -38.902$

4. Берилганлар:  $x = -2.235 \times 10^{-2}$ ,  $y = 2.23$ ,  $z = 15.221$

$$\varphi = \frac{e^{|x-y|} |x-y|^{x+y}}{\arctg x + \arctg z} + \sqrt[3]{x^6 + \ln^2 y}.$$

Натижа:  $\varphi = 39.374$

5. Берилганлар:  $x = 3.981 \times 10^{-2}$ ,  $y = -1.625 \times 10^3$ ,  $z = 0.512$ .

$$a = 2^{-x} \sqrt{x + \sqrt[4]{|y|}} \sqrt[3]{e^{x-1/\sin z}}.$$

Натижа:  $a = 1.26185$

11. Соат милининг сутка бошидаги ҳолати билан  $h$  соат,  $m$  минут ва  $s$  секунддаги ҳолатлари орасидаги  $f$  бурчак (градусларда) аниқлансин ( $0 \leq h \leq 11$ ,  $0 \leq m, s \leq 59$ ).
12. Соат милининг сутка бошига нисбатан  $f$  градус бурилишига мос келувчи  $h$  соат,  $m$  минут ва  $s$  секунд топилсин ( $0 \leq f \leq 360^\circ$ ,  $f$  - ҳақиқий сон).
13. Биринчи январ куни душанба бўлиб, кабиса бўлмаган бирор йилнинг  $k$ -куни ( $1 \leq k \leq 365$ ) ҳафтанинг қайси кунига тўғри келиши аниқлансин ва бу қиймат  $n$  бутун ўзгарувчига ўзлаштирилсин ( $1 \leq n \leq 7$ ).
14. Оғирлиги бир килограмм бўлган маҳсулотнинг нархи берилган. Унинг оғирлиги граммларда киритилсин ва тўлаш зарур бўлган пул миқдори чоп этилсин.
15. 10 метр радиусли цилиндрик шаклга эга бўлган сув босими минорасидаги сув сатҳининг баландлиги берилган бўлса, сувнинг ҳажми ҳисоблансин.
16. Болалар боғчасига бир ойлик тўлов 70000 сўм (бир ой - 22 кун). Агар бола боғчага  $N$  ( $0 < N < 23$ ) кун келмаган бўлса, бир ой учун қанча тўлаш керак бўлади?
17.  $R$  радиусли доирага ички чизилган мунтазам  $n$ -бурчакнинг периметри ва юзаси ҳисоблансин.

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)

2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Тарқатма материаллар

#### 1. Турни бошья турга келтириш кцринишлари

```
Integer_1=(int)Floating;          // ошкор келтириш  
Integer_2=Floating;              // ошкормас келтириш  
static_cast<dataTypeName>(expression)
```

expression – кийматини бошқа турга ўтказиш лозим бўлган ифода.

dataTypeName – ифодани ўтказиш лозим бўлган тур номи.

### Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур киритилган соннинг ASCII жадвалидаги белгисини топиши керак. Лекин дастур соннинг ўзини чоп этмоқда. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

### Тест саволлари

1. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `cout<< "3 / 2 + 5.5 = " << 3 / 2 + 5.5 <<endl;`

- a)  $3 / 2 + 5.5 = 6.5$
- b) 6.0
- c)  $2 + 5.5$
- d) 6.5

2. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `cout<<static_cast<int>(7.9);`

- a) 7
- b) 7.0
- c) 0.9
- d) 7.9

3. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади? `cout<<static_cast<int>(7.8 + static_cast<double>(15 / 2));`

- a) 14
- b) 15
- c) 14.5
- d) 15.3

4. Қуйидаги дастур кодининг натижасини топинг.

`num2 = 4 * 5 - 11; cout<< "num2 = " << num2 <<endl;`

- a) `num2 = 9`
- b)  $4 * 5 - 11 = 9$
- c) `num2 = -24`
- d) `num2 = 19`

5. Қуйидаги дастур кодининг натижасини топинг.

`num2 = 4 / 5 - 11; cout<< num2 <<endl;`

- a) -11
- b) 10.2
- c) -10.2
- d) 11

6. Қуйидаги дастур кодининг натижасини топинг.

```
num2 = 4 % 5 + 5%4; cout<< num2 <<endl;
```

- a) 5
- b) 0
- c) 9
- d) 2.05

7. Қуйидаги дастур кодининг натижасини топинг.

```
num2 = 4 / 5 - 4 % 5; cout<< num2 <<endl;
```

- a) -4
- b) 4
- c) 5
- d) 11

8. Қуйидаги дастур кодининг натижасини топинг.

```
num2 = 4 *3 + 7 / 5 - 25.5; cout<< num2 <<endl;
```

- a) -12.5
- b) 10.5
- c) -12
- d) 12

9. Қуйидаги дастур кодининг натижасини топинг.

```
num = 6; num = num + 2; cout<< num <<endl;
```

- a) 8
- b) 10
- c) -2
- d) 2

10. Қуйидаги ифодаларнинг ичидан хато ифодани аниқланг.

- a)  $q = \% a + b + c + d / 4;$
- b)  $c = (a \% b) * 6;$
- c)  $d = c / b;$
- d)  $e = (a + b + c + d) / 4;$



## Амалий машғулот 5-6.

**Амаллар:** инкремент, декремент, sizeof, мантикий, разрядли, таққослаш

**Ишдан мақсад:** C++ дастурлаш амаллар билан ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда инкремент, декремент, мантикий, разрядли, таққослаш амалларидан фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Берилган уч хонали бутун соннинг рақамлари ўзаро тенг ёки тенг эмаслиги аниқлансин.

Ечиш усули. Масала C++ тилининг бутун сонлар устидаги арифметик амаллардан фойдаланган ҳолда ечилади. Берилган бутун a ва b сонлар учун '/' амали a/b бўлинманинг бутун қисмини, '%' амали a%b бўлинманинг бутун қолдиқини беради. Бу бўлишлардан фойдаланиб, берилган соннинг рақамларини ажратиб олиш ва уларни ўзаро солиштириш мумкин.

Дастур коди:

dastur.cpp файли:

```
#include <iostream>
using namespace std;
int main()
{
    unsigned int n, a2, a1, a0;
    cout << "\nn - qiymatini kiriting: ";
    cin >> n;
    if (n<100 || n>999)
    {
        cout << "Kiritilgan son 3 xonali emas!";
        cout << endl;
        system("pause");
        return 1;
    }
}
```

```

}
a2 = n / 100;
a1 = (n % 100) / 10;
a0 = n % 10;
cout << "Berilgan son raqamlari o'zaro teng";
if (a2 != a1 && a1 != a0 && a2 != a0) cout << " emas!";
else cout << "!";
cout << endl;
system("pause");
}

```

Дастур ишлаши натижасидан намуналар:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
n - qiymatini kiriting: 333
Berilgan son raqamlari o'zaro teng!
Press any key to continue . . .

```

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
n - qiymatini kiriting: 525
Berilgan son raqamlari o'zaro teng!
Press any key to continue . . .

```

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
n - qiymatini kiriting: 3658
Kiritilgan son 3 xonali emas!
Press any key to continue . . .

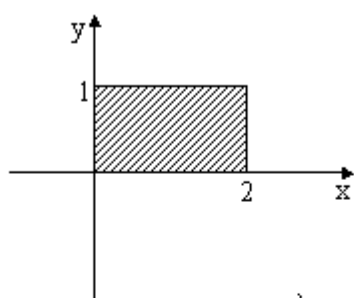
```

### Амалий топшириқлар

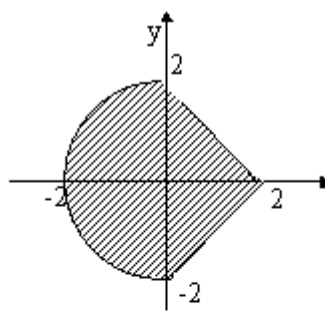
1. Қуйидаги шартлар бажарилганда “*рост*”, акс ҳолда “*ёлгон*” қиймат қабул қилувчи мантикий муносабатлар С++ тилида ёзилсин:
  - a) к бутун сони 7 га бўлинади;
  - b)  $ax^2+bx+c=0$  тенглама ҳақиқий илдизларга эга эмас;
  - d)  $(x,y)$  нукта, маркази  $(1,0)$  нуктада бўлган  $r$  радиусли айлананинг ташқарисида ётади;
  - e)  $n$  натурал сони – тўлиқ квадрат.

2. Қуйидаги шартлар бажарилганда рост, акс ҳолда ёлғон қиймат қабул қилувчи мантиқий муносабатлар C++ тилида ёзилсин:
  - a)  $0 < x < 1$ ;
  - b)  $x = \max(x, y, z)$ ;
  - d)  $x \neq \max(x, y, z)$  (инкор амали қўлланилмасин);
  - e) a, b мантиқий ўзгарувчилардан камида биттаси true;
  - ф) ҳар иккала a, b мантиқий ўзгарувчилар қийматлари true.
3. Ҳисоблансин:
  - a) `false || (1/1 > 0)`; b) `(1/2 > 0) && true`.
4. Қуйидаги шартлар бажарилганда рост, акс ҳолда ёлғон қиймат қабул қилувчи ифодалар C++ тилида ёзилсин:
  - a)  $x \in [0, 1]$ ;
  - b)  $x \notin [0, 1]$ ;
  - d)  $x \in [2, 5] \vee x \in [-1, 1]$ ;
  - e)  $x \notin [2, 5] \vee x \in [-1, 1]$ .
  - f) x, y, z сонларидан ҳар бири мусбат;
  - g) x, y, z сонларидан ҳеч бўлмаганда бири мусбат;
  - h) x, y, z сонларидан ҳеч бири мусбат эмас;
  - i) x, y, z сонларидан фақат бири мусбат;
5. Қуйидаги шартлар бажарилганда рост, акс ҳолда ёлғон қиймат қабул қиладиган ифода C++ тилида ёзилсин:
  - a) бутун n ва m сонлари бир пайтда тоқ ёки жуфт сонлар;
  - b) a, b мантиқий ўзгарувчилардан фақат биттаси true қийматига эга;
  - d) a, b, c мантиқий ўзгарувчилардан фақат биттаси true қийматини қабул қилади.
6. Қуйидаги шарт бажарилганда t мантиқий ўзгарувчиси true, акс ҳолда false қийматини ўзлаштирсин:
  - a) x, y, z сонлар ўзаро тенг;
  - b) x, y, z сонлардан фақат иккитаси ўзаро тенг;
  - c) x мусбат сон;
  - d) p сони q га қолдиқсиз бўлинади (p ва q-натурал сонлар);
  - e)  $ax^2 + bx + c = 0$  тенглама битта ечимга эга, бу ерда a, b ва c ўзгарувчилар 0 бўлиши мумкин;

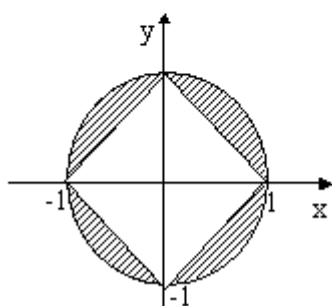
7. Агар  $(x,y)$  нукта бўялган соҳага тегишли бўлса,  $t$  мантикий ўзгарувчи true қийматини қабул қиладиган ифода ёзилсин (расмга қаранг).



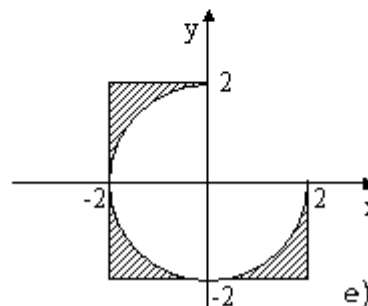
a)



b)



d)



e)

8. Шахмат тахтасининг катаклари  $(x,y)$  бутун сонлар жуфтлиги билан берилади  $(1 \leq x, y \leq 8)$ . Берилган  $(h_1, v_1)$  ва  $(h_2, v_2)$  катаклар учун куйидаги мулоҳазалар мантикий ифода кўринишида ёзлсин:
- катаклари бир хил рангга эга;
  - $(h_1, v_1)$  катагида жойлашган «от»  $(h_2, v_2)$  катакка хавф солади;
  - $(h_1, v_1)$  катагида жойлашган «фарзин»  $(h_2, v_2)$  катакка хавф солади.

### Адабиётлар рўйхати

- [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
- [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
- [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
- Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

1. Инкремент ва декремент амаллари

Префикс инкремент: `++variable`

Постфикс инкремент: `variable++`

Префикс декремент: `--variable`

Постфикс декремент: `variable--`

2. Байт разрядлари устида мантиқий амаллар.

Амаллар	Мазмуни
<code>&amp;</code>	Мантиқий ВА (қўпайтириш)
<code> </code>	Мантиқий ЁКИ (қўшиш)
<code>^</code>	Истисно қилувчи ЁКИ
<code>~</code>	Мантиқий ИНКОР (инверсия)

3. Таққослаш амаллари ва уларнинг қўлланиши.

Амаллар	Қўлланиши	Мазмуни (ўқилиши)
<code>&lt;</code>	<code>a&lt;b</code>	“а кичик b”
<code>&lt;=</code>	<code>a&lt;=b</code>	“а кичик ёки тенг b”
<code>&gt;</code>	<code>a&gt;b</code>	“а катта b”
<code>&gt;=</code>	<code>a&gt;=b</code>	“а катта ёки тенг b”
<code>==</code>	<code>a==b</code>	“а тенг b”
<code>!=</code>	<code>a!=b</code>	“а тенг эмас b”

## Кейслар банки

**Кейс 1.** Разрядли амаллардан фойдаланиб соннинг иккилик кўринишини чоп қиладиган дастур тузилди. Дастур ишга тушурилганда ихтиёрий сонлар учун ишламади.

**Кейсни бажариш босқичлари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тили имкониятларидан дастур ишлайдиган сонларни кенгайтиринг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
cout<< "3 / 2 + 5.5 = " << 3 / 2 + 5.5 <<endl;
```

- a)  $3 / 2 + 5.5 = 6.5$
- b) 6.0
- c)  $2 + 5.5$
- d) 6.5

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
num2 = 4 / 5 - 4 % 5;
```

```
cout<< num2 <<endl;
```

- a) -4
- b) 4
- c) 5
- d) 11

3. С++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
int a=3;
```

```
cout<< a++ <<endl;
```

- a) 3
- b) 2
- c) a++
- d) 4

4. С++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
int a=5;
```

```
int b=a+ (++a);
```

```
cout<< b;
```

- a) 12
- b) 13
- c) 11
- d) 10

5. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
int a=5;
int b=(a--) + (++a);
cout<< ++a;
```

- a) 6
- b) 5
- c) 4
- d) 7

6. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
int a=5, b=2;
b=(a>=b);
cout << b;
```

- a) 1
- b) 0
- c) 5>=2
- d) 2

7. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
int a=5, b=2;
b=(a>=b);
cout << a;
```

- a) 5
- b) true
- c) 5>=2
- d) 2

8. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
bool c;
int a=5, b=2;
c=(a!=b) + (a>b);
cout << c ;
```

- a) 1
- b) 2
- c) 0
- d) 3

9. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
short a=5;  
unsigned int d=123;  
cout << sizeof(a) << ' ' << sizeof(d);
```

- a) 2 4
- b) 4 2
- c) 5 123
- d) 2

10. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
short a=5;  
long d=123;  
cout << sizeof(d);
```

- a) 4
- b) 2 4
- c) 5
- d) 123



## Амалий машғулот 7. Ўқиш-ёзиш оқимлари (cin, cout)

**Ишдан мақсад:** C++ дастурлаш тилида ўқиш ва ёзиш оқимлари билан ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда берилганлар ва қийматларни турли кўринишларда чиқариш ва манипуляторлардан фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масalani C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Экрандан киритилган қиймат оддий, экспоненциал ва фиксирланган кўринишларда чоп қилинсин.

Дастур коди:

dastur.cpp файли:

```
#include <iostream>
using namespace std;
int main()
{
    double qiymat;
    cin >> qiymat;
    cout << "Oddiy ko'rinishda: \n";
    cout << "qiymat = " << qiymat << endl << endl;
    cout << scientific;
    cout << "Eksponensial ko'rinishda: \n";
    cout << "qiymat = " << qiymat << endl << endl;
    cout << fixed;
    cout << "Fiksirlangan ko'rinishda: " << endl;
    cout << "qiymat = " << qiymat << endl << endl;
    cout << endl;
    system("pause");
}
```

Дастур ишлаши натижаси:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
58.25
Oddiy ko'rinishda:
qiymat = 58.25

Ekspontensial ko'rinishda:
qiymat = 5.825000e+001

Fiksirlangan ko'rinishda:
qiymat = 58.250000

Press any key to continue . . . _

```

### Амалий топшириқлар

- $$\gamma = 5 \arctg x - \frac{1}{4} \arccos x \frac{x + 3|x - y| + x^2}{|x - y|z + x^2}.$$
- $$\psi = \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right| + (y - x) \frac{\cos y - \frac{z}{(y - x)}}{1 + (y - x)^2}.$$
- $$b = y^{\sqrt[3]{|x|}} + \cos^3 y \frac{|x - y| \left( 1 + \frac{\sin^2 z}{\sqrt{x + y}} \right)}{e^{|x - y|} + x/2}.$$
- $$f = \frac{\sqrt[4]{y + \sqrt[3]{x - 1}}}{|x - y|(\sin^2 z + tgz)}.$$
- $$h = \frac{x^{y+1} + e^{y-1}}{1 + x|y - tgz|} (1 + |y - x|) + \frac{|y - x|^2}{2} - \frac{|y - x|^3}{3}.$$

6. Массалари мос равишда  $m_1$  ва  $m_2$  бўлган ва бир биридан  $r$  масофада жойлашган иккита жисмнинг ўзаро тортишиш кучи  $F$  аниқлансин.
7. Тўғрибурчакли учбурчакнинг гипотенузаси ва катети берилган бўлса, иккинчи катет ва ички чизилган айлананинг радиуси ҳисоблансин.
8. Айлана узунлиги берилган, бу айлана билан чегараланган доира юзаси топилсин.
9. Учбурчак бурчакларининг катталиклари ва ички чизилган айлана радиуси билан берилган. Учбурчак томонлари узунликлари ҳисоблансин.

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.- М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Тарқатма материаллар.

1. Ўзгарувчига берилиши мумкин бўлган қийматлар жадвали.

Ўзгарувчининг тури	Қабул қилиш қиймати
char	Пробелдан бошқа битта белгини қабул қилади
int	Бутун турдаги ихтиёрий қийматни қабул қилади
double	Ҳақиқий турдаги ихтиёрий қийматларни қабул қилади

## Кейслар банки

**Кейс 1.** Ҳақиқий сонни турли кўринишларда экранга чоп қиладиган дастур тузилди. Энди ҳақиқий соннинг ҳақиқий қисми икки хона аниқликда чоп қилинадиган бўлиши керак.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур коддини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=15, b=5;  
b=b+a%10;  
cout << a+b;
```

- a) 25
- b) 12
- c) 15
- d) 20

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=23, b=4;  
b=a%10;  
cout << a+b;
```

- a) 26
- b) 12
- c) 23
- d) 27

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=10, b;  
b=a%10;  
cout<<b;
```

- a) \*0
- b) 1
- c) 10
- d) 100

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=10, b;  
if (a%2==0) b=a*a;  
else  
b=-a;  
cout<<b;
```

- a) \*100
- b) -10
- c) 10
- d) 0

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=10, b=23;  
switch(a){  
case 1: c=a+b; break;  
case 2: c=a*b; break;  
default: c=(a+b)*b; break;  
}  
cout<<c;
```

- a) 759
- b) 33
- c) 230
- d) хато ҳақида хабар чиқади

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=2,b=4;
if (a!=2-b) a=b;
else b=a;
cout<<a;
```

- a) 4
- b) 5
- c) 2
- d) 8

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=5,b=5, c=9;
if (a!=b || a != c - b) {a=a++; b=--a;}
else a=--a;
cout<<a;
```

- a) 5
- b) 3
- c) 1
- d) 7

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 0, b = 10;
if (pow(b,a) == 1)
    a = b * (b++);
else
    a = b * (b--);
cout<<a;
```

- a) 100
- b) 81
- c) 121
- d) 18

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 0, b = 1;
if (++a == b)
    a += a;
else
    a += b;
cout<<a;
```

- a) 2
- b) 8
- c) 3
- d) 4

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 11, b = 22;
if (abs(a - b) == a) a *= a;
else a *= b;
cout<<a;
```

- a) 121
- b) 100
- c) 81
- d) 64

## Амалий машғулот 8-9.

### Операторлар. Шарт операторлари

**Ишдан мақсад:** C++ дастурлаш тилида шарт операторлари билан ишлаш, унинг турли кўринишларидан фойдаланиш кўникмаларига эга бўлиш. Дастурда `?:`, `if`, `if...else`, `switch` шарт операторларидан фойдалана олиш.

**Масаланинг қўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Ҳақиқий турдаги  $x_1, y_1, x_2, y_2, x_3, y_3$  сонлари берилган. Координата маркази учлари  $(x_1, y_1), (x_2, y_2)$  ва  $(x_3, y_3)$  нукталарда бўлган учбурчак ичида ётадимми?

Ечиш усули. Бирорта  $(x, y)$  нукта учбурчак ичига тегишли бўлади, агар бу нуктанинг учбурчак учлари билан ҳосил қилувчи учбурчаклари юзаларининг  $(s_1, s_2$  ва  $s_3)$  йиғиндиси шу учбурчак юзасига  $(s)$  тенг бўлса:  $s = s_1 + s_2 + s_3$ . Шунини қайд этиш керакки, иккита ҳақиқий турдаги қийматларнинг ўзаро тенглигини текшириш улар айирмасининг абсолют қиймати берилган аниқликдан кичиклиги орқали аниқланади.

Учбурчакнинг юзаси Герон формуласи ёрдамида топилади:

$$s = \sqrt{p(p-a)(p-b)(p-c)},$$

бу ерда  $a, b, c$  – учбурчак томонларининг узунликлари,  $p$  – учбурчак периметрининг ярми.

Дастур коди:

dastur.cpp файли:

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
float x1, y1, x2, y2, x3, y3;
```



```

float a, b, c;
float s1, s2, s3;
float s;
float p;
const float eps = 0.0001; // tenglik aniqligi
cout << "\n1 - nuqta koordinatasini kiriting (x1, y1): ";
cin >> x1 >> y1;
cout << "\n2 - nuqta koordinatasini kiriting (x2, y2): ";
cin >> x2 >> y2;
cout << "\n3 - nuqta koordinatasini kiriting (x3, y3): ";
cin >> x3 >> y3;
//{(x1,y1),(x2,y2),(x3,y3)} uchburchak yuzasini
hisoblash(s)
a = sqrt((x1 - x2)*(x1 - x2) + (y1 - y2)*(y1 - y2));
b = sqrt((x2 - x3)*(x2 - x3) + (y2 - y3)*(y2 - y3));
c = sqrt(pow(x1 - x3, 2) + pow(y1 - y3, 2));
p = (a + b + c) / 2;
s = sqrt(p*(p - a)*(p - b)*(p - c));
//{(0,0),(x1,y1),(x3,y3)} uchburchak yuzasini hisoblash
(s1)
a = sqrt(x1*x1 + y1*y1);
b = sqrt(x3*x3 + y3*y3);
p = (a + b + c) / 2;
s1 = sqrt(p*(p - a)*(p - b)*(p - c));
//{(0,0),(x2,y2),(x3,y3)} uchburchak yuzasini hisoblash
(s2)
a = sqrt(x2*x2 + y2*y2);
c = sqrt(pow(x2 - x3, 2) + pow(y2 - y3, 2));
p = (a + b + c) / 2;
s2 = sqrt(p*(p - a)*(p - b)*(p - c));
//{(0,0),(x1,y1),(x2,y2)} uchburchak yuzasini hisoblash
(s3)
b = sqrt(x1*x1 + y1*y1);
c = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));

```

```

p = (a + b + c) / 2;
s3 = sqrt(p*(p - a)*(p - b)*(p - c));
// s=s1+s2+s3 shartni tekshirish. Bunda tenglikka
tekshirish
// qiymatlarni ayirmasining absolyut qiymatini nolga
// yaqinligi bilan almashtiriladi, chunki haqiqiy
sonlarni
// ustida amallar bajarilganda aniqlik yo'qotilishi
mumkin
if (fabs(s - (s1 + s2 + s3))<eps)
    cout << "Koordinata markazi uchburchak ichida.";
else
    cout << "Koordinata markazi uchburchak ichida emas.";
cout << endl;
system("pause");
}

```

Дастур ишлаши натижаси:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
1 - nuqta koordinatasini kiriting (x1, y1): 2 4
2 - nuqta koordinatasini kiriting (x2, y2): 1 5
3 - nuqta koordinatasini kiriting (x3, y3): -3 5
Koordinata markazi uchburchak ichida emas.
Press any key to continue . . .

```

### Амалий топшириқлар

1. Ҳақиқий  $x, y$  ва  $z$  сонлари берилган бўлиб,  $x < y < z$  муносабат ўринли бўлса бу сонлар иккилантирилсин, акс ҳолда бу сонлар абсолют қийматлари билан алмаштирилсин.

2. Учта ихтиёрий  $a, b$  ва  $c$  сон берилган. Томонларининг узунликлари шу сонларга тенг бўлган учбурчак мавжудми?
3. Сонлар ўқида учта  $A, B$  ва  $C$  нуқталар жойлашган.  $B$  ва  $C$  нуқталардан қайси бири  $A$  нуқтага яқин масофада жойлашган бўлса, шу масофа чоп этилсин.
4. Берилган уч хонали сон рақамлари орасида бир хиллари бор ёки йўқлиги аниқлансин?
5. Агар томонларининг узунликлари ихтиёрий  $a, b$  ва  $c$  сонларга тенг бўлган учбурчакни куриш мумкин бўлмаса  $0$ , акс ҳолда – учбурчак тенг томонли бўлса  $3$ , тенг ёнли бўлса  $2$  ва бошқа ҳоллар учун  $1$  қийматини чоп қилувчи программа тузилсин.
6. Агар учта ҳақиқий, ўзаро тенг бўлмаган  $x, y$  ва  $z$  сонлар йиғиндиси  $1$  дан кичик бўлса, учта соннинг энг кичиги қолганлари йиғиндисининг ярмиси билан алмаштирилсин, акс ҳолда  $x$  ва  $y$  лардан кичиги қолганлари йиғиндисининг ярми билан алмаштирилсин.
7. Бутун турдаги  $a, b$  ва  $c$  ўзгарувчилар қиймати шундай алмаштирилсинки, натижада  $a \leq b \leq c$  муносабат ўринли бўлсин.
8. Натурал  $n (n \leq 9999)$  сони берилган. Сонни тўрт хонали деб ҳисобга олган ҳолда ушбу соннинг палиндром эканлиги аниқлансин (чапдан ва ўнгдан бир хил ўқиладиган сонлар, масалан,  $1221, 5555, 440$  сонлари палиндром сонлар ҳисобланади).
9. Берилган тўрт хонали соннинг бошидаги иккита рақамлари йиғиндиси қолган рақамлари йиғиндисига тенг ёки йўқлиги аниқлансин.
10. Берилган ҳақиқий мусбат сон каср қисмининг бошидаги учта рақамлари орасида  $0$  рақами борми?
11. Эски япон календаридида  $60$  йиллик такрорланиш қабул қилинган ва бу такрорланиш ўз навбатида бешта  $12$  йиллик такрорланиш остиларидан (қисмларидан) иборат. Қисм такрорланишлар қуйидаги рангларнинг номи билан белгиланган: яшил, қизил, сариқ, оқ ва қора. Ҳар бир такрорланиш остининг ичидаги йиллар ҳайвонларнинг номи билан белгиланган: сичқон, сигир, йўлбарс, қуён, аждарҳо, илон, от, қўй, маймун, товук, ит ва тўнғиз ( $1984$ -йил – яшил сичқон йили – кейинги такрорланишнинг боши бўлган). Ерамизнинг бирор йили киритилиб, унинг эски япон календаридидаги номи чоп қилинсин.

## Адабиётлар рўйхати

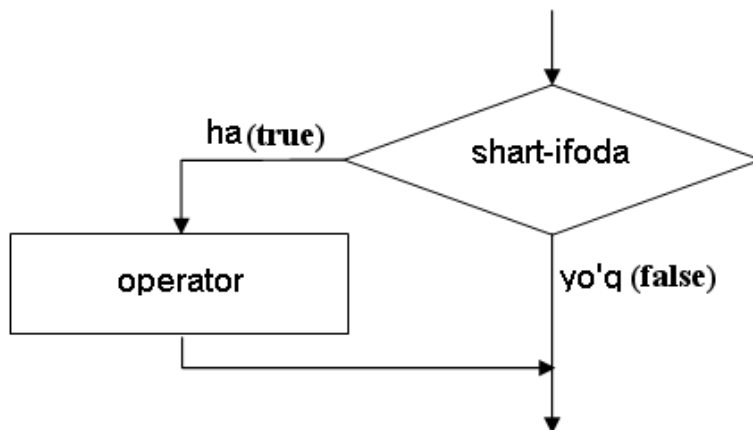
1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

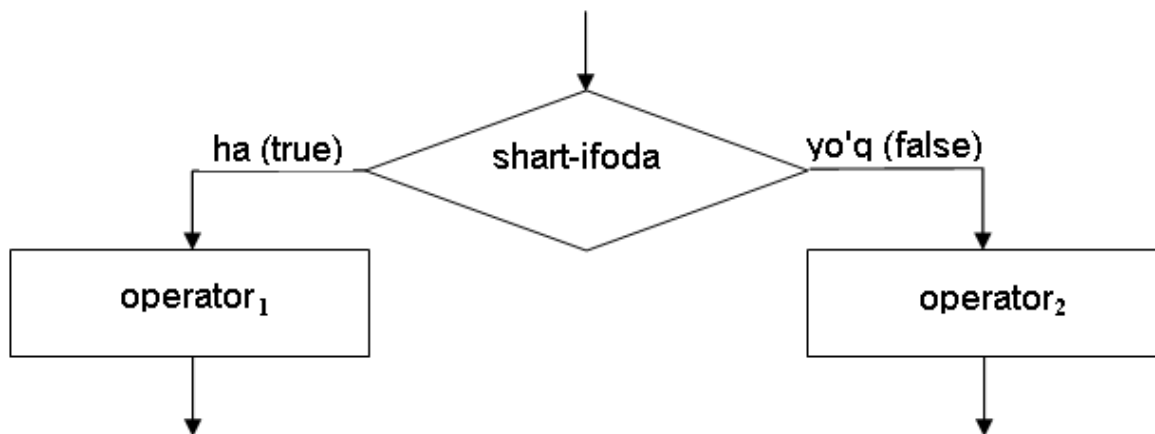
### 1. Мантикий операторлар.

Оператор	Белги	Намуна
ВА	&&	ифода1 && ифода2
ЁКИ		ифода1    ифода2
ИНКОР	!	!ифода

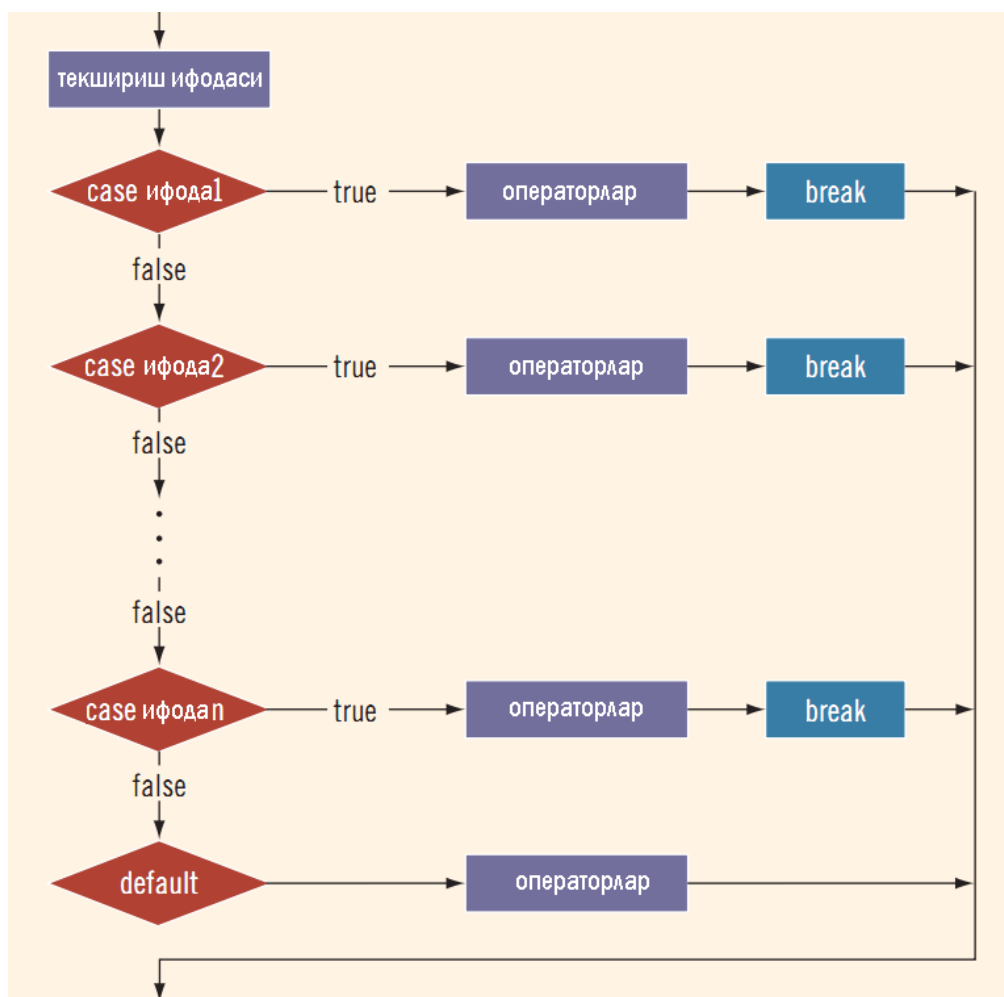
### 2. if шарт ифодаси блок-схемаси.



3. `if...else` шарт операторининг бажарилиши.



4. `switch` шарт операторининг бажарилиши.



## Кейслар банки

**Кейс 1.** `if` шарт оператори ёрдамида экранда танланган давлатни қайси қитъага тегишлилигини аниқловчи дастур тузилди. Танланиши мумкин бўлган давлат номларини кўпайтириш керак. Аммо дастурдаги танланиши мумкин бўлган давлатлар номи кўпайтирилса `if` конструкцияси кодлари ҳам ҳар бир даслат учун алоҳида ёзилиши керак. Натижада дастур коди кўпайиб кетади.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 16, b = 5;  
if (sqrt(a) == b) a *= ++a;  
else a *= ++b;  
cout<<a;
```

- a) 96
- b) 15
- c) 56
- d) 19

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 3, b = 3;  
if (pow(a++, b) == 65) a += b*4;  
else a *= ++b;  
cout<<a;
```

- a) 16
- b) 15

c) 26

d) 19

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 3, b = -1;  
if (pow(b,a) == 1) a = a / (b++);  
else a = b / (b--);  
cout<<a;
```

a) 1

b) 5

c) 2

d) 9

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 2, b = a + 3;  
if (b - b / a != 1/2)  
    a = b * a;  
else  
    a = b / a;  
cout<<a;
```

a) 10

b) 15

c) 20

d) 19

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 9, b = 10;  
if (b * a == 1)  
    a = b * b;  
else  
    a = b / a;  
cout<<a;
```

- a) 2
- b) 1
- c) 2
- d) 4

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 0, b = 1;
```

```
if (a == b)
```

```
    a += a;
```

```
else
```

```
    a += b;
```

```
cout<<a;
```

- a) 7
- b) 1
- c) 0
- d) 5

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 5, b = 3;
```

```
if (a - b == b) a += a;
```

```
if(a % 5 == 0) a += b * b;
```

```
else a += b;
```

```
cout<<a;
```

- a) 37
- b) 15
- c) 14
- d) 17

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 17, b = 18;
```

```
if (a == b) a += 9;
```

```
else a = a - 3;
```



```
if(a % 2 == 0) a += b * b;
else a += b;
cout<<a;
```

- a) 150
- b) 250
- c) 338
- d) 418

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 7, b = 8;
if (a == b) a = a - a;
else if(a % 2 == 0) a = b * b;
else if(a % 2 != 0) a += b;
cout<<a;
```

- a) 91
- b) 18
- c) 15
- d) 22

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 26, b = 5;
if (sqrt(a) == b) a = sqrt(a);
else a = b + a;
cout<<a;
```

- a) 30
- b) 31
- c) 25
- d) 35

## Амалий машғулот 10-11.

### Такрорлаш операторлари. Бошқарувни узатиш операторлари

**Ишдан мақсад:** C++ дастурлаш тилида цикллар, такрорланувчи жараёнлар билан ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда турли кўринишдаги такрорлаш операторларидан, бошқарувни узатиш операторларидан фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Фибоначчи сонлари қйидагича аниқланади:

$f_0 = f_1 = 1, f_n = f_{n-1} + f_{n-2}, n = 2, 3, \dots$  Фибоначчи сонларидан ҳосил бўлган кетма-кетликнинг  $n$  та ҳади топилсин. Дастур кодлари турли шарт ва бошқарувни узатиш операторлари (for, while, do-while ва goto операторлари) ёрдамида тузилсин.

Дастур коди:

dastur\_for.cpp файли:

```
#include<iostream>
using namespace std;

int main(){
    int n, f0, f1, f2, i;
    cout << "n = ";
    cin >> n;
    f0 = f1 = 1;
    cout << f0 << " " << f1;
    for (i = 2; i <= n; i++)
    {
        f2 = f0 + f1;
        f0 = f1;
```

```

        f1 = f2;
        cout << " " << f2;
    }
    system("pause");
    return 0;
}
Дастур ишлаши натижаси:

```

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
n = ?
1 1 2 3 5 8 13 21
Press any key to continue . . . _

```

Dastur\_while.cpp файли:

```

#include<iostream>
using namespace std;

int main(){
    int n, f0, f1, f2, i;
    cin >> n;
    f0=f1=1;
    cout << f0 << " " << f1;
    i=2;
    while (i<=n)
    {
        f2=f0+f1;

```

```

    f0=f1;
    f1=f2;
    cout << " " << f2;
    i++;
}
system("pause");
return 0;
}

```

### Амалий топшириқлар

1. Берилган  $n$  та ҳақиқий сонлар орасида кўшниларидан (ўзидан олдинги ва кейинги сонлардан) катта бўлган сонлар миқдори топилсин.
2. Берилган 10 та натурал сонларнинг энг катта умумий бўлувчиси топилсин.
3. Берилган  $n$  ва  $m$  натурал сонлари учун  $s = \sum_{i=1}^n \prod_{j=5}^m (i + j)$  ифоданинг қиймати ҳисоблансин.
4. Берилган  $n$  на натурал сонлардан иборат кетма-кетликнинг тартиб номерлари Фибоначчи сонлари бўлган ҳадларининг йиғиндиси ҳисоблансин.
5. 0 дан 15 гача бўлган сонлар иккилик кўринишида чоп қилинсин.
6. Берилган натурал  $n$  ва  $m$  учун  $s = \prod_{i=3}^n \sum_{j=2}^m (2 * i + j * j)$  ифоданинг қиймати ҳисоблансин.
7. Берилган натурал  $n$ ,  $m$  ва ҳақиқий  $a$  сонлари учун  $s = \sum_{i=4}^n \sum_{j=2}^m (a^i + j)$  ифоданинг қиймати ҳисоблансин.
8. 0 билан тугайдиган сонлар кетма-кетлиги берилган (0 сони кетма-кетликка кирмайди). Кетма-кетликдаги барча мусбат сонлар йиғиндиси топилсин.
9. Натурал  $n$  сони ва  $n$  та ҳақиқий сонлардан иборат кетма-кетлик берилган. Тоқ ўринда турган сонлар махимуми ва жуфт ўриндагиларнинг минимуми топилсин.
10. Берилган  $n$  натурал сондаги турли рақамлар миқдори аниқлансин.
11. Натурал  $n$  сони ва  $n$  та сонлар жуфтлиги берилган -  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$ . Ҳар бир сонлар жуфтлиги текисликдаги нуқта координатаси деб ҳисоблаб, берилган нуқталарни ўз ичига олувчи, маркази координата бошида

бўлган энг кичик айлананинг радиуси топилсин.

12. Рақамлари йиғиндиси  $n$  сонига тенг ( $1 \leq n \leq 27$ ) бўлган уч хонали натурал сонлар сони -  $k$  топилсин. Бутун сонларни бўлиш амалларидан ( $/$ ,  $\%$ ) фойдаланилмасин.
13. Ўнлик ёзувида бир хил рақамлари бўлмаган уч хонали сонлар ўсиш тартибида чоп этилсин (бутун сонларни бўлиш амалидан фойдаланилмасин).
14. Тақрибий усулда  $\int_0^{3.14} \ln(2 + \sin(x)) dx$  интеграл тўғри тўртбурчаклар формуласидан фойдаланган ҳолда ҳисоблансин:

$$\int_a^b f(x) dx \approx h[f(x_1) + \dots + f(x_n)],$$

бу ерда  $h = \frac{b-a}{n}$ ,  $x_i = ih - h/2$ ,  $i=1..n$ ,  $n=100$ .

15. Бутун  $n$  ( $n > 1$ ) сони ва  $n$  та ҳақиқий сонлардан иборат кетма-кетлик берилган. Кетма-кетлик ўсувчи ёки йўқлиги аниқлансин.
16. Бутун  $n$  ( $n > 1$ ) сони ва  $n$  та ҳақиқий сонлардан иборат кетма-кетлик берилган. Кетма-кетликнинг манфий элементлари орасидан энг каттаси топилсин.
17. Стандарт функциялардан фойдаланмаган ҳолда ( $\text{fabs}()$ -бундан мустасно) берилган  $\varepsilon > 0$  аниқликда й қиймати ҳисоблансин. Йиғиндини ҳисоблашда навбатдаги қўшилувчи ҳад модули бўйича  $\varepsilon$  дан кичик бўлган ҳолда ҳисоблаш жараёни тўхтатилсин.

a)  $y = \sin x = x + x^3/3! + x^5/5! + \dots + x^{2n+1}/(2n+1)! + \dots;$

b)  $y = \cos x = 1 - x^2/2! + x^4/4! - \dots + (-1)^n x^{2n}/(2n)! + \dots;$

c)  $y = \ln(1+x) = x - x^2/2 + x^3/3 - \dots + (-1)^{n-1} x^n/n + \dots, (|x| < 1);$

d)  $y = \arctg x = x - x^3/3 + x^5/5 - \dots + (-1)^n x^{2n+1}/(2n+1) + \dots, (|x| < 1).$

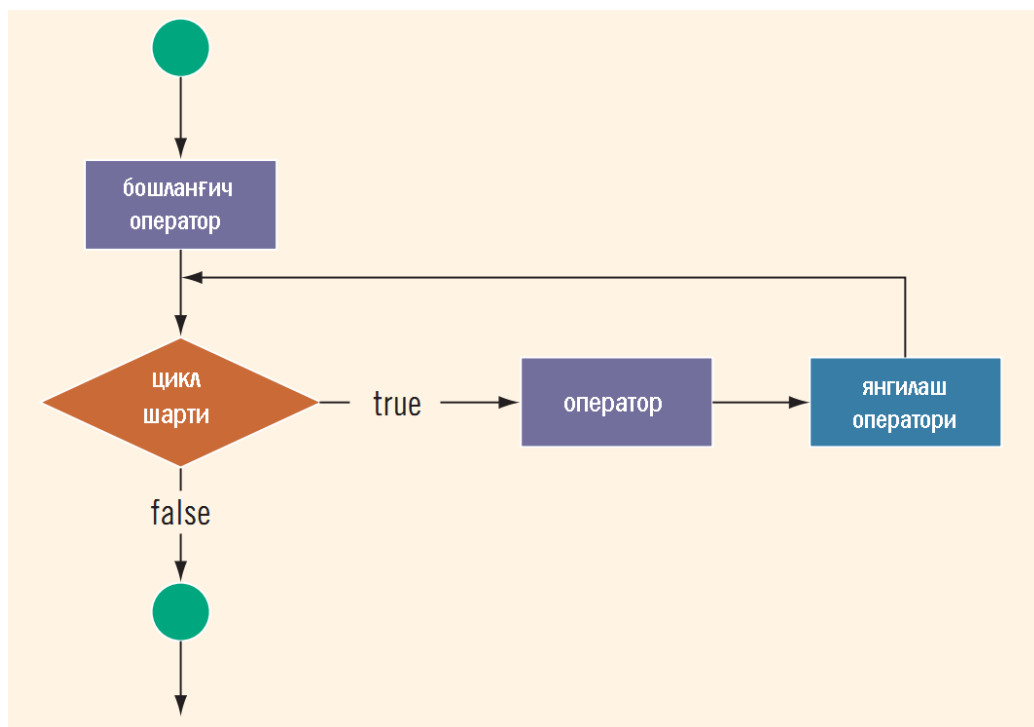
### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)

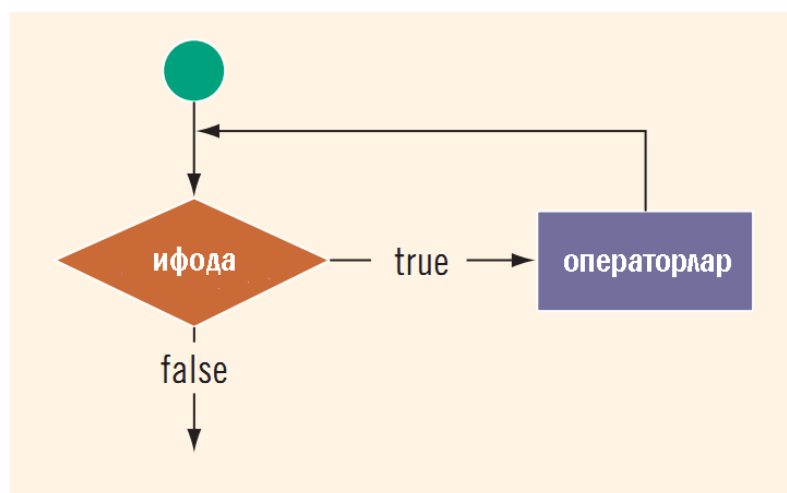
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Тарқатма материаллар.

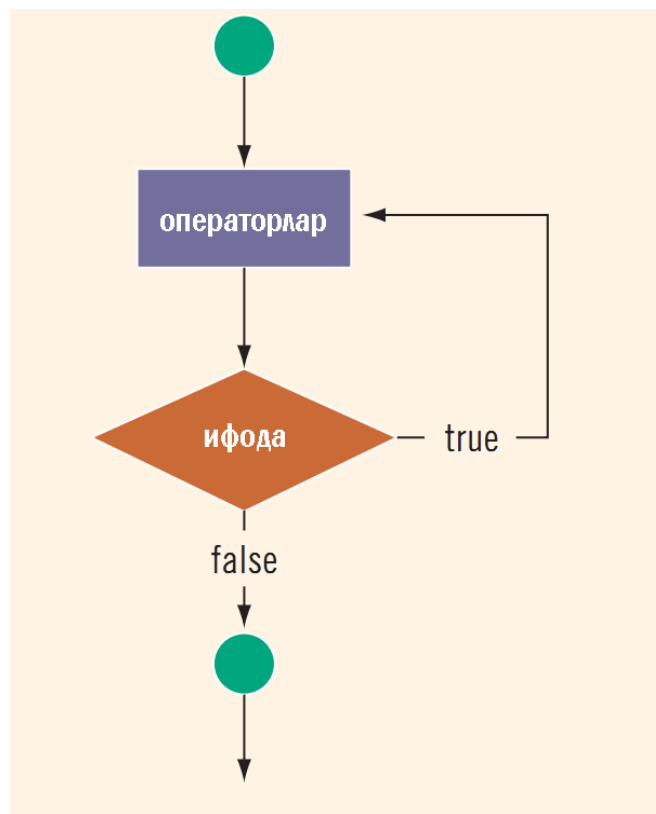
#### 1. for такрорлаш оператори ишлаш схемаси



#### 2. while такрорлаш оператори ишлаш схемаси.



### 3. do-while такрорлаш оператори ишлаш схемаси.



### Кейслар банки

**Кейс 1.** Цикллар билан ишлайдиган дастур тузилди. Дастур вазифаси киритилган соннинг туб бўлувчиларини чоп қилишдан иборат. Дастур ишлаши натижасида фақат биринчи туб бўлувчисигина чоп этилмоқда.

#### Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 0;
for (int i = 1; i < 100; i*=4)
{
    if (i / 12 <= 5) a += i;
    if (i > 98) a = a - i;
}
cout<<a;
```

- a) 16
- b) 85
- c) 56
- d) 108

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 13, a = 0, b = 22;
while (i <= b)
{
    i+=3;
    a+=i;
}
cout<<a;
```

- a) 82
- b) 150
- c) 56
- d) 91



3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 2, a = 0, b = 12;
a = a + b;
do
{
    i+=3;
    a=i%10;
}
while (i <= b);
cout<<a;
```

- a) 16
- b) 15
- c) 4
- d) 1

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 0;
for (int i = 0; i < 50; i+=5)
{
    if (i / 10 >= 3)
        if (i % 10 != 5)
            a += i;
}
cout<<a;
```

- a) 70
- b) 75
- c) 85
- d) 80

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 2, a = 0, b = 0;
while (i <= b)
{
    i = i + 2;
    a+=i%10;
    b = b + 3/2;
}
cout<<a;
```

- a) 0
- b) 3
- c) 2
- d) 1

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 2, a = 2, b = 19;
do {
    i+=5;
    a+=i%5;
}
while (i <= b);
cout<<a;
```

- a) 15
- b) 12
- c) 10
- d) 85

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=10,b=0,l=0;
```

```
do
```

```
{
```

```
  b++;
```

```
  l+=b;
```

```
}
```

```
while(b<10);
```

```
cout<<l;
```

a) 37

b) 35

c) 55

d) 77

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=0;
```

```
for(int b=0; a<5; b++)
```

```
  a+=b;
```

```
  cout<<a;
```

a) 6

b) 5

c) 2

d) 12

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=105,b=10,i=0,s=0;
```

```
while(i<10)
```

```
{ i++;
```

```
  a=a-b;
```

```
  s=s+a%b;
```

```
}
```

```
cout<<s+i*a;
```

- a) 90
- b) 180
- c) 100
- d) 122

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=0;
for(int b=0; b<5; b++);
a=a+1;
cout<<a+1;
```

- a) 2
- b) 3
- c) 5
- d) 4

## Амалий машғулот 12-13.

### Статик массивлар

**Ишдан мақсад:** C++ дастурлаш тилида статик массивлар билан ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда бир ўлчамли ва кўп ўлчамли статик массивлардан фойдалана олиш.

**Масаланинг қўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Берилган  $s$  векторнинг қийматлари иккиннинг даражаларига (1,2,4,8,16,...) тенг бўлган элементларнинг индекслари ва уларнинг йиғиндиси чоп қилинсин.

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;

int main(){
    const int n=10;
    int s[n], k;
    cout << "Massiv elementlari soni = " << n << "\n";
    for (int i = 0; i < n; i++)
    {
        cout << "s[" << i + 1 << "]=";
        cin >> s[i];
    }
    k = 0;
    cout << "Indekslar: ";
    for (int i = 0; i < n; i++)
    {
        bool b = false;
        unsigned long int d = 1;
```

```

    for (;;)
    {
        if (s[i] == d)
        {
            k = k + i;
            cout << i << " ";
            break;
        }
        d = d * 2;
    }
}
cout << "Indekslar yig'indisi = " << k << endl;
system("pause");
return 0;
}
Дастур ишлаши натижаси:

```

The screenshot shows a debug console window with the following output:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
Massiv elementlari soni = 10
s[1]=4
s[2]=2
s[3]=7
s[4]=10
s[5]=64
s[6]=31
s[7]=28
s[8]=121
s[9]=128
s[10]=42
Indekslar: 0 1 4 8
Indekslar yig'indisi = 13
Press any key to continue . . .

```

### Амалий топшириқлар

1. Натурал  $n$  сони ва  $a_1, a_2, \dots, a_n$  бутун сонлардан иборат кетма-кетлик берилган. Кетма-кетликда бир марта қатнашган элементлар чоп қилинсин.
2. Натурал  $n$  сони ва  $a_1, a_2, \dots, a_n$ ,  $b_1, b_2, \dots, b_n$  сонлар кетма-кетликлари берилган. Бу кетма-кетликлар бир-биридан фақат элементларининг

жойлашув тартиби билангина фарқ қилиш ёки йўқлиги аниқлансин.

3. Натурал  $n$  сони,  $A = \{a_1, a_2, \dots, a_n\}$  ва  $B = \{b_1, b_2, \dots, b_n\}$  сонлар кетма-кетликлари берилган.  $A$  кетма-кетлик  $[1..n]$  оралигидаги ҳар хил бутун сонлардан ташкил топган (индекслар).  $B$  кетма-кетликдаги элементлар  $A$  кетма-кетликда кўрсатилган жойларга ўтказилсин, яъни  $b_i$  элемент  $a_i$  индекс бўйича жойлашсин.
4. Жуфт бўлган  $n$  натурал сони ва ярмигача қийматлар билан тўлдирилган  $A = \{a_1, a_2, \dots, a_n\}$  кетма-кетлик берилган. Кетма-кетлик қийматли элементларини такрорлаб суриш орқали тўлдирилсин (масалан,  $A = \{3, 8, \dots\}$  учун  $A = \{3, 3, 8, 8, \dots\}$ ).
5. Натурал  $n$  сони ва  $A = \{a_1, a_2, \dots, a_n\}$  кетма-кетлик берилган. Кетма-кетликдаги элементлар ўсиш тартибида бўлган энг узун қисм кетма-кетлик топилсин.
6. Берилган  $x, y, z$  векторларни элемент туридаги объектлар тўплами деб қараб (агарда  $k$  элемент  $x$  тўпламга тегишли бўлса  $x[k]=true$ , акс ҳолда  $x[k]=false$  ва ҳоказо), ушбу вектор - тўпламлар устида қуйидагилар амалга оширилсин:
  - а) агар  $x$  тўплам  $y$  тўпламнинг қисм тўплами бўлса,  $y$  ҳолда  $t$  ўзгарувчига  $true$  қиймати, акс ҳолда  $false$  берилсин;
  - б)  $z = x \cap y$  - тўпламлар кесишмаси топилсин;
  - д)  $z = x \cup y$  - тўпламларнинг бирлашмаси топилсин;
  - е)  $z = x \setminus y$  - тўпламларнинг айирмаси топилсин ( $z$  тўпламга  $x$  тўпламнинг  $y$  тўпламга кирмаган барча элементлари кирази).
7. 

```
const int n = 20;
float B[n][n];
```

Қўшимча матрисадан фойдаланмаган ҳолда  $B$  матрицанинг транспонерланган кўриниши БТ ҳосил қилинсин.
8.  $n$  натурал сони ва 5-чи тартибли ҳақиқий турдаги квадрат матрица берилган. Бу матрицанинг  $n$ -даражаси топилсин ( $A^1=A, A^2=AA, A^3=A^2A$  ва ҳоказо).
9. 

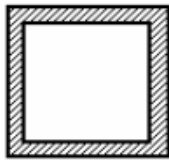
```
const int n=20;
float nuqta[n][2], d;
```

Матрицанинг сатр элементларини текисликдаги нуқталарнинг

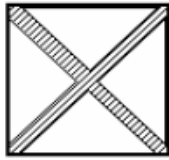
координаталари деб қараб, шу нуқталар орасидаги энг катта масофа топилсин.

10. float A[9][9],s;

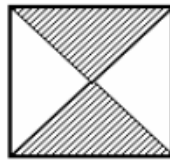
A матритсанинг қуйидаги расмдаги бўялган соҳалардаги элементлари йиғиндиси с топилсин.



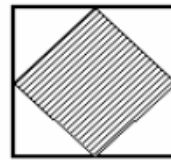
a)



b)



c)



d)

11. int A[10][10],B[9][9];

int n,k; //  $0 \leq n \leq 10; 0 \leq k \leq 10;$

Берилган A матритсанинг n-сатри ва k-устунини ўчириш орқали B матритса ҳосил қилинсин.

12. сонст инт n=8, m=12;

инт k, C[n][m];

C матритсанинг "махсус" элементлар сони - k аниқлансин.

Элемент "махсус" дейилади, агар у ўзи жойлашган устундаги бошқа элементлар йиғиндисидан катта ва ўзи жойлашган сатрда чапдаги элементлардан катта, ўнгдагиларидан эса кичик бўлса.

13. Матритсанинг элементи эгар нуқта дейилади, агарда у бир вақтнинг ўзида шу элемент жойлашган сатрдаги энг кичиги ва устундаги энг каттаси бўлса ёки аксинча. 10 x 15 ўлчамли бутун турдаги матритсанинг эгар нуқтаси индекси чоп этилсин.

14. Элементлари бир-бирига тенг бўлмаган хақиқий турдаги 7-тартибли квадрат матритса берилган. Энг катта элементи жойлашган сатрнинг энг кичик элементи жойлашган устунга кўпайтмаси топилсин.

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)





- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

### Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={ 15,25,45,20,45,30};
int b[]={ 35,25,5,30,5,20};
int s=0,n=6;
for(int i=0;i<6;i++) s+=a[i]+b[n-1-i];
cout<<s;
```

- a) 160
- b) 300
- c) 320
- d) 350

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=3;
int a[n][n]={{ 7,2,3},{ 3,2,4},{ 0,6,4}};
int s=0;
for(int i=0;i<n;i++)
    s+=a[0][i]+a[i][n-1];
cout<<s;
```

- a) 16
- b) 23
- c) 32
- d) 35

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={ 15,30,25,10,45,20};
int s=0;
for(int i=1;i<5;i++)
    if(a[i]>a[i-1]&& a[i]>a[i+1])
        s+=a[i];
cout<<s;
```

- a) 75
- b) 30
- c) 80
- d) 85

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={35,28,37,22,11,17};
int max=a[0];
for(int i=1;i<6;i++)
    if(a[i]>max) max=a[i];
cout<<max;
```

- a) 16
- b) 37
- c) 32
- d) 35

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=3;
int a[n][n]={{7,2,8},{3,2,4},{4,6,9}};
int s=0;
for(int i=0;i<n;i++)
    for(int j=0;j<n;j++)
        s+=a[i][j]; cout<<s;
```

- a) 23
- b) 31
- c) 45
- d) 55

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={ 15,30,25,10,45,20};  
    int s=0;  
for(int i=1;i<5;i++) if(a[i]>a[i-1]&&a[i]>a[i+1]) s+=a[i];  
    cout<<s;
```

- a) 75
- b) 125
- c) 110
- d) 85

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={ 35,28,37,22,11,17};  
    int max=a[0];  
    for(int i=1;i<6;i++)  
        if(a[i]>max) max=a[i];  
    cout<<max;
```

- a) 37
- b) 35
- c) 11
- d) 17

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=4;  
int a[n][n], s=0;  
for(int i=0;i<n;i++)  
    { a[i][0]=1;  
      for(int j=1;j<n;j++) a[i][j]=2*a[i][j-1];  
    }  
for(int i=0;i<n;i++) s+=a[i][i];  
cout<<s;
```

- a) 15
- b) 25
- c) 20
- d) 18

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=3;
int a[n][n]={ {7,2,8},{3,2,4},{4,6,9} };
int s=0;
for(int i=0;i<n;i++) s+=a[n/2][i];
cout<<s;
```

- a) 9
- b) 18
- c) 13
- d) 2

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=4;
int a[n][n], s=0;
for(int j=0;j<n;j++)
{ a[0][j]=1;
  for(int i=1;i<n;i++) a[i][j]=2*a[i-1][j];
}
for(int i=0;i<n;i++) s+=a[i][i]+a[i][n-1-i];
cout<<s;
```

- a) 30
- b) 20
- c) 25
- d) 35

## Амалий машғулот 14-15.

### Функциялар эълон қилиш ва аниқлаш. main() функцияси. Локал ва глобал ўзгарувчилар

**Ишдан мақсад:** C++ дастурлаш тилида функциялар билан ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда турли кўринишдаги функциялар, локал ва глобал ўзгарувчилардан фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Берилган бутун соннинг рақамлари йиғиндиси ҳисоблансин.

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;

int Raqamlar_Summasi(int);
int main()
{
    int b;
    cout << "\nButun sonni kiriting: ";
    cin >> b;
    cout << "\n" << Raqamlar_Summasi(b);
    cout << endl;
    system("pause");
    return 0;
}
int Raqamlar_Summasi(int n)
{
    int s=0;
    if (n < 0) n =- n;
    while (n > 0)
```

```

    {
        s += n % 10;
        n /= 10;
    }
    return s;
}

```

Дастур ишлаши натижаси:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
Butun sonni kiriting: 36985
31
Press any key to continue . . .

```

### Амалий топшириқлар

1. Ихтиёрий  $n$  та соннинг йиғиндисини ҳисоблаш функцияси тузилсин.
2. Ихтиёрий  $n$  та соннинг максимумини топиш функцияси тузилсин.
3. Иккита учбурчак учларининг координаталари билан берилган. Бу учбурчаклардан қайси бирининг юзаси катта эканлигини аниқловчи функция тузилсин.
4.  $n$  натурал сони ва  $n$  та сонлардан иборат кетма-кетлик берилган. Кетма-кетликдаги ишора алмашилишлар сонини аниқлаш функцияси тузилсин.
5. Берилган  $n$  ва  $m$  ( $n \geq m$ ) натурал сонлар учун Нютон биноми коэффисиентларини ҳисоблаш функцияси тузилсин: 
$$C_n^m = \frac{n!}{(n-m)!m!}.$$
6.  $a_1 + ib_1$  ва  $a_2 + ib_2$  комплекс сонлар устида арифметик амаллар ('+', '-', '\*', '/') бажарувчи функция тузилсин, бу ерда  $a_1, b_1, a_2, b_2$  - берилган бутун сонлар.

7. Текисликда  $a, b, c$  ва  $d$  кесмалар берилган. Уларнинг ҳар бир учтасидан учбурчак қуриш мумкинлигини аниқлайдиган мантиқий  $Uchburchak()$  функция тузилсин. Агар  $Uchburchak()$  функцияси бирорта учлик учун true қиймат қайтарса, шу учликдан ҳосил бўлган учбурчакнинг юзасини ҳисобловчи ҳақиқий турдаги  $Uchburchak\_Yuzasi()$  функцияси тузилсин.
8. Бутун  $a, b, p, q$  ( $b \neq 0$ ) параметрига эга  $qisqart(a,b,p,q)$  функция  $a/b$  касрни қисқармайдиган  $p/q$  кўринишига олиб келсин ва ундан  $1+1/2+1/3+\dots+1/20$  касрни қисқармайдиган  $c/d$  кўринишига келтиришда фойдаланилсин.
9. Учбурчак  $a, b$  ва  $c$  томонлари билан берилган. Учбурчакнинг медианаларидан иборат учбурчакнинг медианалари топилсин. (Изоҳ: учбурчакнинг  $a$  томонига ўтказилган медиана  $0.5\sqrt{2b^2 + 2c^2 - a^2}$  га тенг).
10. Ҳақиқий  $c, d$  ( $c < d$ ) сонлари берилган. Қуйидаги интеграл ҳисоблансин:

$$\int_c^d \arctg^2 x dx + \int_0^{\pi} \sin e^{10x} dx .$$

Биринчи интеграл  $n=20$  да, иккинчиси  $n=100$  да трапетсия формуласидан фойдаланиб ҳисоблансин:

$$\int_a^b f(x) dx \approx h \cdot \left[ f(a)/2 + \sum_{i=1}^{n-1} f(a+ih) + f(b)/2 \right]$$

бу ерда  $h = (b - a) / n$ .

11. Берилган  $1/(1+x^2) = x, 3e^x + x = 0$  ва  $x * \ln(1+x) = 0.5$  тенгламаларнинг ечимлари  $\varepsilon > 0$  аниқликда топилиб, ўсиш тартибида чоп этилсин.
12. Учбурчак учлари ва шу учбурчак ичидаги бирор нуқта координатаси билан берилган. Берилган нуқтадан учбурчак томонларигача бўлган энг яқин масофа топилсин.
13. Текисликда учта тўғри чизик  $a_k x + b_k y = c_k$  ( $k = 1, 2, 3$ ) тенгламалари билан берилган. Агар бу тўғри чизиклар жуфт-жуфти билан кесишиб, учбурчак ҳосил қилса, шу учбурчак юзаси ҳисоблансин.
14.  $n$  дан  $2n$  ( $n > 2$ ) гача бўлган сонлар ичидан “егизаклар” жуфти чоп этилсин. (Иккита туб сонлар ”егизак” дейилади, агарда улар бир-бири билан 2 га фарқ қилса, масалан: 41 ва 43 сонлари).



## Адабиётлар рўйхати

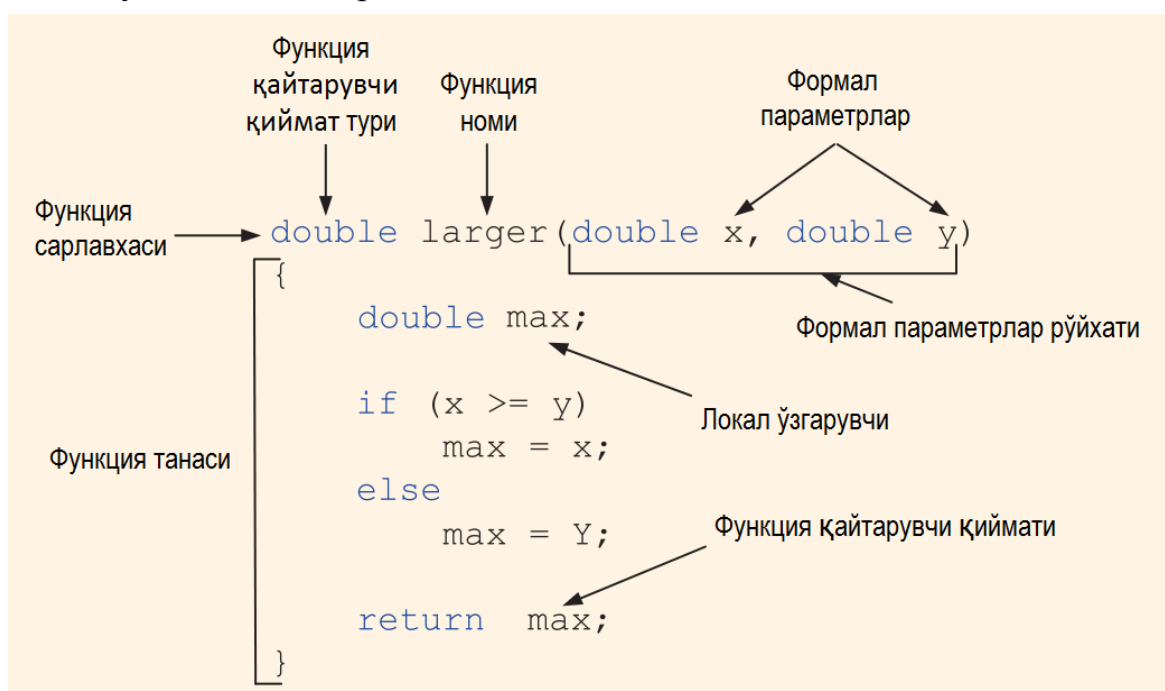
1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

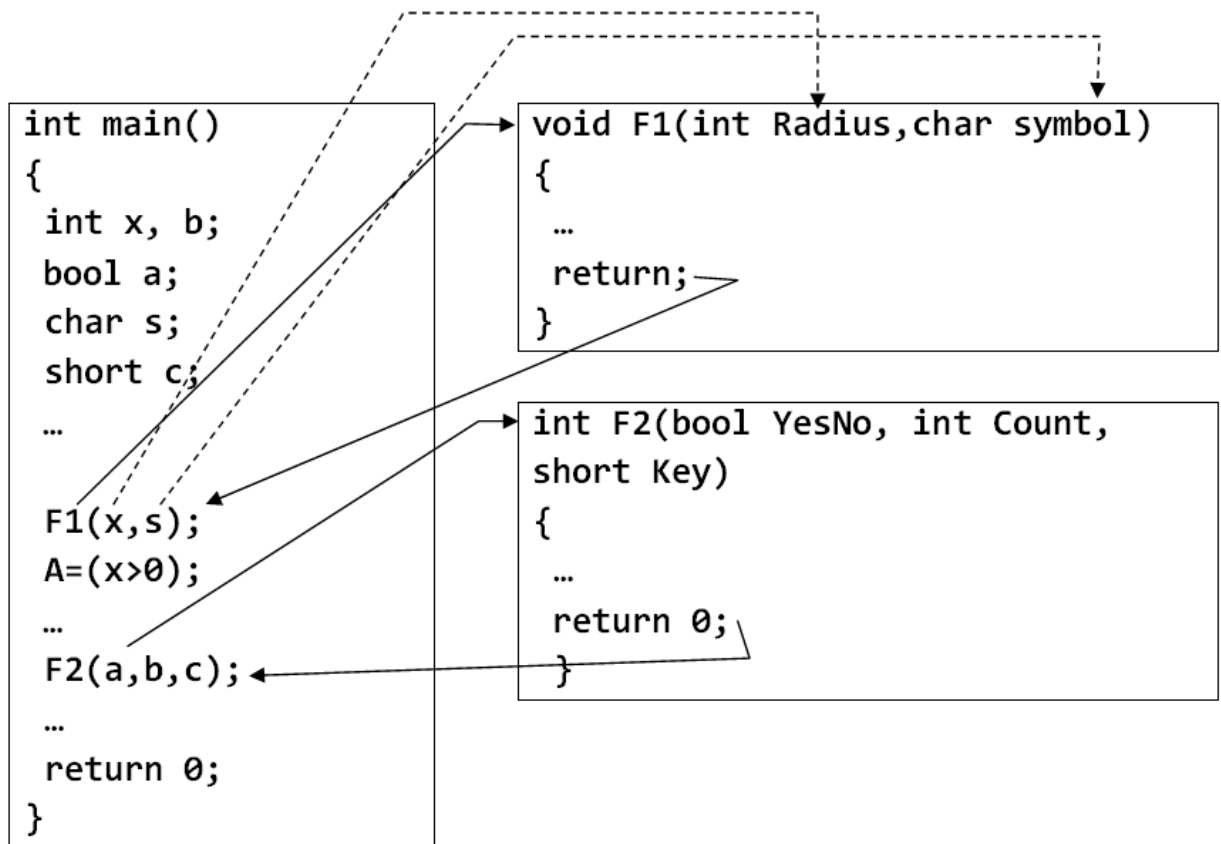
1. Кўп ишлатиладиган математик функциялар жадвали:

<code>abs(x)</code>	<code>&lt;cmath&gt;</code>	x бутун сонининг абсолют қийматини қайтаради
<code>fabs(x)</code>	<code>&lt;cmath&gt;</code>	x ҳақиқий сонининг абсолют қийматини қайтаради
<code>log(x)</code>	<code>&lt;cmath&gt;</code>	x сонининг натурал логарифмини қайтаради
<code>pow(x, y)</code>	<code>&lt;cmath&gt;</code>	$x^y$ ҳисоблайди
<code>sqrt(x)</code>	<code>&lt;cmath&gt;</code>	x сонининг квадрат илдизини қайтаради

2. Функция қисмлари.



### 3. Функция ишлатилиши.



### Кейслар банки

**Кейс 1.** Функциялар билан ишловчи дастур тузилди. Дастур вазифаси берилган интегрални эпсилон аниқликда ҳисоблаш. Лекин дастур ишлаши натижасида эпсилон аниқликда ҳисоблашга эришилмади.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
double SUM(double *a, int n)
{
    double t=0;
    t=t+a[0];
    for (int i=1;i<n;i++)t=t+a[i]/a[i-1];
    return t;
}
void main()
{
    double a[]={7,1,3,5,3},c;
    c=SUM(a,3);
    cout << c;
}
```

- a) 10
- b) 15
- c) 20
- d) 30

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int n, int k=5)
{
    k=n/k;
    return n-k;
}
void main()
{
    int n=7, a[]={1,2,3,6,4}, c=2, k=2;
    for (int i=0;i<5;i++)
    if (F(n,a[i]-n)-F(n-a[i],c)<0) c=c+F(c, c);
    cout << c;
}
```

- a) 5
- b) 7
- c) 8
- d) 6

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int n, int k, int t)
{
for (int i=0; t; )
    {t--;n=n+k;}
return n;
}
void main()
{
int a[]={2,5,2,6,7}, n=2, k=1, t=3;
for (int i=0;i<3;i++)
if (2*F(a[i], a[i], a[0])>t) t=t+F(a[i],t,a[0]);
cout << t;
}
```

- a) 51
- b) 17
- c) 38
- d) 26

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int a)
{
a=5;
return a;
}
void main()
{
int a=7;
```

```
F(a)+F(a+5);
```

```
cout << a;
```

```
}
```

a) 9

b) 7

c) 11

d) 10

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int n, int k=5)
```

```
{
```

```
k=n-k;
```

```
return n-k;
```

```
}
```

```
void main()
```

```
{
```

```
int n=7, a[]={6,2,3,8,1}, c=5, k=2;
```

```
for (int i=0;i<5;i++)
```

```
if (F(c,a[i]-n)-F(n-a[i],c)<0) c=c+F(c, c);
```

```
cout << c;
```

```
}
```

a) 230

b) 160

c) 210

d) 180

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int a)
```

```
{
```

```
return a+9;
```

```
}
```

```
void main()
```

```
{
```

```
int a=7;
int t = F(a)+F(a+5);
cout << t;
}
```

- a) 37
- b) 35
- c) 16
- d) 15

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int &n, int k=3)
{
k=n-k;
return n-k;
}
void main()
{
int n=7, a[]={5,6,2,4,3}, c=0, k=2;
for (int i=0;i<5;i++)
if (F(n-5)) c=c+F(a[i], k);
cout << c;
}
```

- a) 37
- b) 25
- c) 10
- d) 15

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int &n, int k=3)
{
k=n-k;
return n-k;
}
```

```

void main()
{
int n=7, a[]={6,2,3,8,1}, c=5, k=2;
for (int i=0;i<5;i++)
if (F(n-c,c)) c=c+F(a[i], c);
cout << c;
}

```

- a) 150
- b) 250
- c) 160
- d) 180

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```

int F(int a)
{
return a*9;
}
void main()
{
int a=2;
int t = a+F(a+1);
cout << t;
}

```

- a) 29
- b) 18
- c) 31
- d) 27

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```

int F(int n, int k=3, int t=13)
{
n=5;
for (int i=0; i<=k;i++)

```

```
    t=t+n;
return n;
}
void main()
{
int a[]={3,6,5,7,9,1}, n=a[0], k=a[1], t=a[3];
for (int i=0;i<6;i++)
if (F(a[i])>a[i]) t=t+F(a[i],a[2],a[0]);
cout << t;
}
```

- a) 19
- b) 20
- c) 17
- d) 18



## Амалий машғулот 16.

### Рекурсив функциялар

**Ишдан мақсад:** C++ дастурлаш тилида рекурсив функциялар билан ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда турли кўринишдаги ва қийинчиликдаги рекурсив функциялардан фойдалана олиш.

**Масаланинг қўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Қуйидаги

$$x^n = \begin{cases} 1, & \text{агар } n = 0; \\ 1/x^{|n|}, & \text{агар } n < 0; \\ x \cdot x^{n-1}, & \text{акс холда.} \end{cases}$$

формула ёрдамида  $x^n$  катталикини ҳисобловчи  $\text{Daraja}(x,n)$  рекурсив функция тузилсин. Бу ерда  $x$  ( $x \neq 0$ ) ҳақиқий,  $n$ -бутун сон.

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;

double daraja(double x,int n)
{
    if (!n) return 1;
    else if (n > 0) return x * daraja(x, n - 1);
    else return daraja(x, n + 1) / x;
}

int main()
{
    double x;
    int n;
```

```

cout << "x=";
cin>>x;
cout << "n=";
cin>>n;
cout << "x^n=" << daraja(x,n);
system("pause");
return 0;
}

```

Дастур ишлаши натижаси:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
x = 5
n = 3
x^n = 125
Press any key to continue . . . _

```

### Амалий топшириқлар

1. 5 та ҳар хил натурал сон берилган. Бу сонларни барча ўрин алмашиш ҳолатлари чоп қилинсин.
2. Шахмат тахтачасида 8 та фарзин шундай жойлаштирилган-ки, улар бир-бирини “урмайди“. Барча шундай ҳолатлар (92 та) чоп этилсин.
3. Берилган номанфий  $n$  ва  $m$  бутун сонлар учун  $A(n,m)$  функция қиймати ҳисоблансин.

$$A(n, m) = \begin{cases} m + 1, & n = 0; \\ A(n - 1, 1), & n \neq 0, m = 0; \\ A(n - 1, A(n, m - 1)), & n > 0, m > 0. \end{cases}$$

4. Бутун  $n$  ва  $n$  та ҳар хил элементдан иборат бўлган ҳақиқий турдаги вектор берилган. Қуйидаги тез тартибланиш усули ёрдамида массив элементлари ўсиш тартибида жойлаштирилсин: массивнинг ихтиёрий элементини танлаб, (масалан ўртадагисини) шу элементнинг чап томонида шу элементдан кичик, ўнг томонда эса катта элементлари жойлаштирилади. (Бу билан танланган элемент ўзининг охирига жойига киради), кейин шу усул массивнинг чап ва ўнг қисми учун рекурсив қўлланилади.
5. 1 дан  $n$  гача номерланган  $n$  та аҳоли пункти мавжуд. Айрим пунктлар ўзаро йўллар билан туташган. Бу йўллар ёрдамида 1-пунктдан  $n$ -пунктга бориш мумкин ёки йўқлигини аниқлансин. Йўллар ҳақидаги маълумот  $i$  ва  $j$  ( $i < j$ ) сонлар жуфтликлари кетма-кетлик кўринишида берилган. Кетма-кетлик 2 та нол билан тугайди.
6. Кесмани тенг иккига бўлиш усули ёрдамида  $f(x)=0$  тенгламани  $[a,b]$  ораликда эпс аниқлигидаги илдизини топадиган  $\text{root}(f,a,b,\text{eps})$  рекурсив функцияси тузилсин. ( $\text{eps}>0, a<b, f(a)*f(b)<0, [a,b]$  ораликда  $f(x)$  – узлуксиз ва монотон функция деб ҳисоблансин.)
7.  $C_n^0 = C_n^n = 1; C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$  формула бўйича  $C_n^m$  бином коэффисентини ҳисоблайдиган  $C(n,m)$  рекурсив функция тузилсин, бунда  $0 \leq m \leq n$ .
8. `const int n=40;`  
`float x[n];`  
Берилган  $x$  векторнинг  $k$  ва охириги ўриндаги элементлари орасидаги энг кичик элементни топадиган  $\text{min1}(k)$  ёрдамчи рекурсив функция қурилиб, унинг ёрдамида  $x$  векторининг энг кичик элементини топадиган  $\text{min}(x)$  функцияси тузилсин.

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

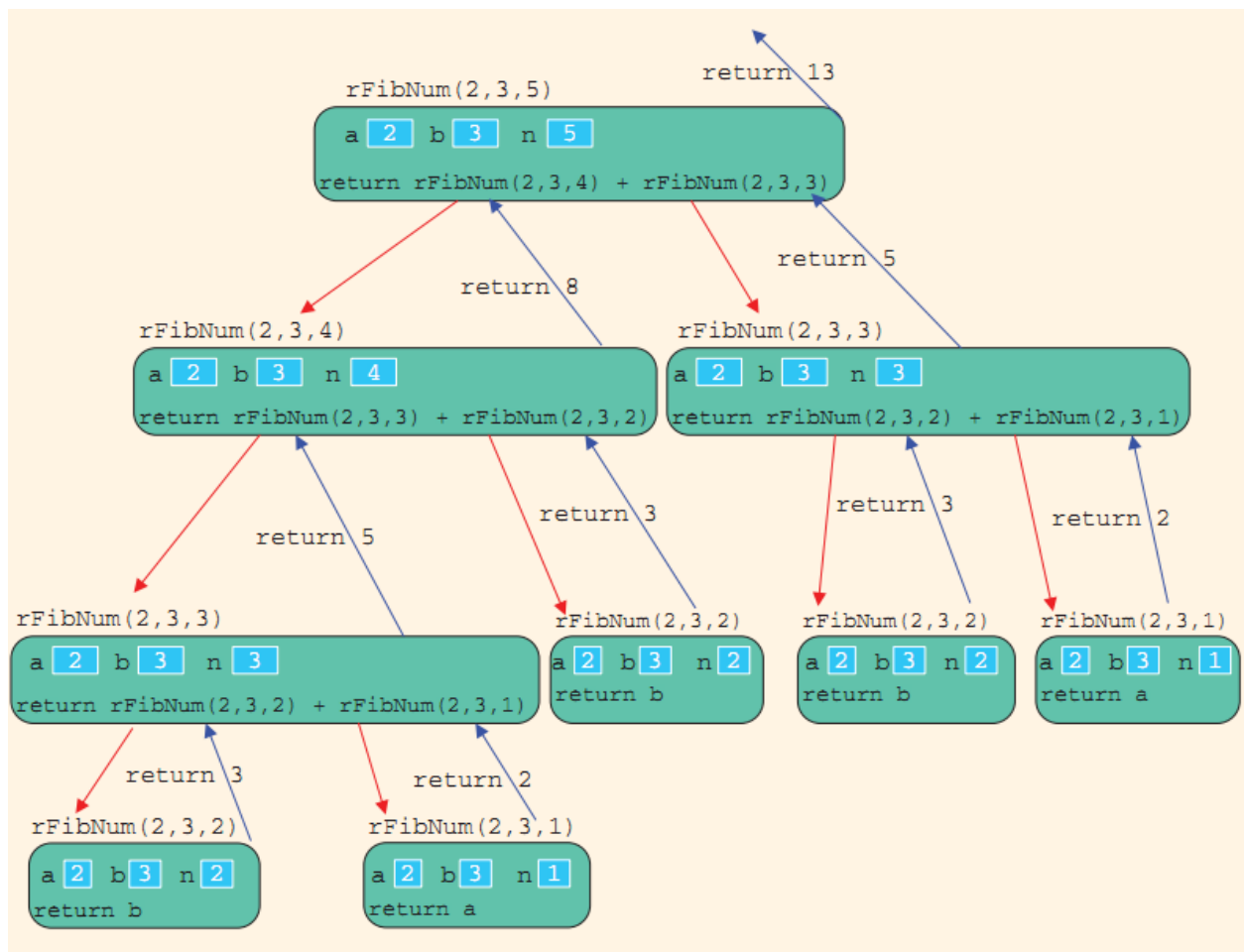
## Тарқатма материаллар.

### 1. Факториал рекурсив функциясининг ишлаш схемаси

↓	$F(4)=4 \cdot F(3)$	↓	$F(4)=4 \cdot F(3)$	↓	$F(4)=4 \cdot F(3)$	↓	$F(4)=4 \cdot F(3)$	↑	$F(4)=4 \cdot 6$
↓	$F(3)=3 \cdot F(2)$	↓	$F(3)=3 \cdot F(2)$	↓	$F(3)=3 \cdot F(2)$	↑	$F(3)=3 \cdot 2$		
↓	$F(2)=2 \cdot F(1)$	↓	$F(2)=2 \cdot F(1)$	↑	$F(2)=2 \cdot 1$				
↓	$F(1)=1 \cdot F(0)$	↑	$F(1)=1 \cdot 1$						
↑	$F(0)=1$								

4! ҳисоблаш схемаси

### 2. Фибоначчи сонларини топувчи рекурсив функциянинг ишлаш схемаси.



## Кейслар банки

**Кейс 1.** Рекурсив функциялар билан ишлайдиган дастур тузилди. Дастур вазифаси киритилган соннинг факториалини топиш. Дастур ишлаши давомида натижа чиқмади, хотира тўлиб кетгани ҳақида тизим хатоси хабари чиқди. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тилидада дастур кодини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int n, int k=9, int t=13)
{
int i=5;
for (i=0; i!=0; )
    i--;
return n+i-k+t;
}
void main()
{
int a[]={2,7,6,3,7}, n=a[0], k=a[1], t=a[3];
for (int i=0;i<3;i++)
if (2*F(a[i], a[i])) t=t+F(a[i],t,a[0]);
cout << t;
}
```

- a) 8
- b) 9
- c) 6
- d) 1

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=3,b=2;
while(a%2==1)
{
a++;
if (b) b--;
a=a+b*3;
}
cout << a;
```

- a) 8
- b) 10
- c) 7
- d) 9

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int n, int k, int t)
{
int i=5;
for (i=0; i!=0; )
i--;
return n+i-k+t;
}
void main()
{
int a[]={2,5,2,6,7}, n=2, k=1, t=3;
for (int i=0;i<3;i++)
if (2*F(a[i], a[i], a[0])>t) t=t+F(a[i],t,a[0]);
cout << t;
}
```

- a) 4
- b) 5
- c) 6
- d) 7

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=4;
int a[n][n];
for(int i=0;i<n;i++)
    for(int j=0;j<n;j++) a[i][j]=1;
int s=0;
for(int i=0;i<n;i++) s+=a[i][i]+a[i][n-1-i];
cout<<s;
```

- a) 8
- b) 9
- c) 7
- d) 6

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=3;
int a[n][n]={{7,2,3},{3,2,4},{0,6,4}};
int s=0;
for(int i=0;i<n;i++)
    s+=a[0][i]+a[i][n-1];
cout<<s;
```

- a) 23
- b) 31
- c) 21
- d) 11

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=1,b=1,c=5;
while(b>c);
{
    a=a*3+b*c;
    b=abs(b-c);
}
cout << a;
```

- a) 18
- b) 8
- c) 11
- d) 15

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={35,28,37,22,11,17};
    int max=a[0];
    for(int i=1;i<6;i++)
        if(a[i]>max) max=a[i];
    cout<<max;
```

- a) 37
- b) 35
- c) 11
- d) 17

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int n, int k=2, int t=6)
{
    n=9;t=13;k=n+t;
    return 5;
}
int FF(int &n, int &k, int t)
{
    k=n;n=t;t=k;
    return n+t/k+1;
}
void main()
{
    int a[]={4,2,2,2,2}, n=2, k=2, t=2;
    for (int i=0;i<5;i++)
        if (2*F(a[i], a[i], a[0])<2 || FF(k, t, n)) t=t+F(a[i],k);
    cout << t;
}
```



- a) 15
- b) 7
- c) 10
- d) 18

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=3;
int a[n][n]={{7,2,8},{3,2,4},{4,6,9}};
int s=0;
for(int i=0;i<n;i++) s+=a[n/2][i];
cout<<s;
```

- a) 9
- b) 18
- c) 13
- d) 2

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int n=2, int k=3)
{
return n-k;
}
void main()
{
int n=7, a[]={5,6,2,4,2,4,2,7}, c=0;
for (int i=0;i<n;i++)
if (F(a[0]-a[i])) c=c+F(a[i], a[i+1]);
cout << c;
}
```

- a) 7
- b) 10
- c) 9
- d) 13

## Амалий машғулот 17.

### Фойдаланувчи томонидан аниқланган берилганлар турлари

**Ишдан мақсад:** C++ дастурлаш тилида фойдаланувчи томонидан аниқланган берилганлар турлари билан ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда санаб ўтилувчи турлар, typedef орқали аниқланадиган турлардан фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Берилган  $r$  бирликда берилган  $x$  ўзгарувчисининг қиймати метрларда чоп қилинсин.

Дастур коди:

dastur.cpp файли:

```
#include <iostream>
using namespace std;
int main()
{
    enum Birlik { detsimetr, kilometr, metr, millimetr,
santimetr };
    float x, y;
    int t;
    cout << "Uzunlikni kiriting: x=";
    cin >> x;
    cout << " Uzunlik birliklari\n";
    cout << " 0- detsimetr\n";
    cout << " 1- kilometr\n";
    cout << " 2- metr\n";
    cout << " 3- millimetr\n";
    cout << " 4- santimetr\n";
    cout << " Uzunlikni birligini tanlang: t=";
    cin >> t;
```

```

switch (t)
{
    case detsimetr: y = x / 10; break;
    case kilometr: y = x * 1000; break;
    case metr: y = x; break;
    case millimetr: y = x / 1000; break;
    case santimetr: y = x / 100; break;
    default:
        cout << "Uzunlik birligi noto'g'ri kiritildi!";
        return 0;
}
cout << y << " metr";
system("pause");return 0;
}

```

Дастур ишлаши натижаси:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
Uzunlikni kiriting: x=120
Uzunlik birliklari
0- detsimetr
1- kilometr
2- metr
3- millimetr
4- santimetr
Uzunlikni birligini tanlang: t=2
120 metr
Press any key to continue . . . _

```

### Амалий топшириқлар

1. Берилганлар тури ва ўзгарувчилар қуйидагича аниқланган:

```
enum Oy{ yan,fev,mar,apr,may,iyn,iyl,avg,sen,okt,noy,dek };
```

```
int d1,d2; Oy m1,m2; bool t;
```

Агар d1, m1 сана (йил ҳисобида) d2, m2 санадан олдин келса, t ўзгарувчига true, акс ҳолда false қиймат берилсин.

2. Оу  $m, m1$ ; (Ой тури юқоридаги масалада берилган).  
 $\text{int } k, n$ ;  
 Қуйидаги шарт бўйича  $m1$  ўзгарувчига қиймат берилсин:  
 а)  $m$  ойдан кейинги ойнинг номи (декабрдан кейин январ келишини ҳисобга олган ҳолда);  
 б)  $m$  ойдан кейинги  $k$ -ойнинг номи;  
 д) йилнинг  $n$  - ой номи берилсин.
3.  $\text{enum Nota}\{\text{do, re, mi, fa, sol, la, si}\};$   
 $\text{enum Oraliq}\{\text{secund, tersia, qvart, kvint, sekst, septima}\};$   
 $\text{Nota } n1, n2; \text{Oraliq } i;$   
 Берилган  $n1$  ва  $n2$  ( $n1 \neq n2$ ) ноталардан ташкил топган  $i$ -оралиқ аниқлансин;  
 $\text{secund}$  - бу иккита қўшни (айлана бўйлаб) ноталардан ташкил топган оралиқ (масалан,  $\text{re}$  ва  $\text{mi}$ ,  $\text{si}$  ва  $\text{do}$ ),  $\text{tersia}$  – бу битта нотадан кейинги оралиқ (масалан,  $\text{fa}$  ва  $\text{la}$ ,  $\text{si}$  ва  $\text{re}$ ) ва ҳоказо.
4.  $\text{enum Mavsum}\{\text{qish, bahor, yoz, kuz}\};$   
 Оу  $m$ ;  $\text{Mavsum } s$ ; (Ой тури 1-масалада аниқланган).  
 Берилган  $m$  ойга мос келувчи  $s$  мавсум аниқлансин.
5.  $\text{enum Davlat}\{\text{Germaniya, Quba, Laos, Monaco, Nepal, Polsha}\};$   
 $\text{enum Qita}\{\text{Osiyo, Amerika, Evropa}\};$   
 $\text{Davlat } davlat; \text{Qita } qita;$   
 Давлатнинг номи бўйича у жойлашган қитъа номи аниқлансин.
6. Берилган, 7 сўмдан катта бўлган ҳар қандай тийинсиз пул миқдорини 3 ва 5 сўмликлар йиғиндиси билан қайтимсиз тўлаш мумкинлиги исботлансин, яъни берилган  $n > 7$  учун  $3a + 5b = n$  шартни қаноатлантирувчи мусбат бутун  $a$  ва  $b$  сонлар жуфтликлари топилсин.
7. Берилган 10 та натурал сонларнинг энг катта умумий бўлувчиси топилсин.
8. Берилган натурал  $n$  сонининг барча бўлувчилари топилсин.
9. Берилган  $n$  учун  $s$  нинг қиймати ҳисоблансин:  
 а)  $s = 1! - 2! + 3! - 4! + \dots + (-1)^{n+1} n!$ ;  
 б)  $s = -2! + 4! + \dots + (-1)^n (2n)!$ .
10. Кетма-кетлик қуйидаги қонуният билан берилган:
- $$y_0 = 0; \quad y_k = \frac{y_{k-1} + 1}{y_{k-1} + 2}, \quad k = 1, 2, \dots$$
- Берилган  $\varepsilon > 0$  учун  $y_n - y_{n-1} < \varepsilon$  шартни қаноатлантирувчи биринчи  $y_n$  ҳад топилсин.
11. Берилган натурал  $n$  сони рақамларининг йиғиндиси ҳисоблансин.

## Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

1. Hafta санаб ўтилувчи тур кўринишлари

```
enum Hafta {dush, sesh, chor, paysh, juma, shanba, yaksh};  
enum Hafta {dush=8, sesh, chor=12, paysh=13, juma=16, shanba,  
yaksh=20};
```

2. typedef ёрдамида массив учун янги тур яратиш.

```
typedef char Ism[30];  
Ism ism;      // Ism - yangi tur
```

## Кейслар банки

**Кейс 1.** Санаб ўтилувчи тур билан ишлайдиган дастур тузилди. Дастур вазифаси танланган давлат номига нисбатан унинг қитъасини аниқлашдан иборат. Дастурни ишга тушуриб, бирор давлат танланганда давлат қитъаси сифатида барча қитъаларни чоп қилмоқда. Дастурнинг тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).

- C++ дастурлаш тили имкониятларидан келиб чиқиб тўғри натижа беривчи дастур кодини тақдим этинг (индивидуал ҳолда).

### Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
cout<< "3 / 2 + 5.5 = " << 3 / 2 + 5.5 <<endl;
```

- a)  $3 / 2 + 5.5 = 6.5$
- b) 6.0
- c)  $2 + 5.5$
- d) 6.5

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
num2 = 4 / 5 - 4 % 5;
```

```
cout<< num2 <<endl;
```

- a) -4
- b) 4
- c) 5
- d) 11

3. C++ да қуйидаги операторлар кетма-кетлиги бажариниши натижасида экранга нима чиқади?

```
int a=3;
```

```
cout<< a++ <<endl;
```

- a) 3
- b) 2
- c) a++
- d) 4

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 0;
```

```
for (int i = 0; i < 50; i+=5)
```

```
{
```

```
    if (i / 10 >= 3)
        if (i % 10 != 5)
            a += i;
    }
    cout<<a;
```

- a) 70
- b) 75
- c) 85
- d) 80

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 2, a = 0, b = 0;
while (i <= b)
{
    i = i + 2;
    a+=i%10;
    b = b + 3/2;
}
cout<<a;
```

- a) 0
- b) 3
- c) 2
- d) 1

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 2, a = 2, b = 19;
do {
    i+=5;
    a+=i%5;
}
while (i <= b);
cout<<a;
```

- a) 15
- b) 12
- c) 10
- d) 85

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a=10,b=0,l=0;
do
{
    b++;
    l+=b;
}
while(b<10);
cout<<l;
```

- a) 37
- b) 35
- c) 55
- d) 77

8. Қуйидаги дастур қисми бажарилиши натижасида экранга нима чиқади?

```
enum Hafta {dush=8, sesh, chor, paysh, juma, shanba, yaksh};
enum Oy {yanvar=8, fevral, mart, aprel, may};
Hafta a;
Oy b;
a=juma;
b=may;
cout << a+b;
```

- a) 24
- b) 6
- c) 7
- d) 12



9. Қуйидаги дастур қисмидаги хато кодни топинг.

```
enum Hafta {dush=8, sesh, chor, paysh, juma, shanba, yaksh};
```

```
Hafta a;
```

```
a=juma;
```

```
a++;
```

a) a++;

b) dush=8

c) Hafta a;

d) a=juma;

10. Дастур қисми охирида popularSport ўзгарувчиси қайси константани олади.

```
enum sports {BASKETBALL, FOOTBALL, HOCKEY, BASEBALL,  
SOCCER,VOLLEYBALL};
```

```
sports popularSport = FOOTBALL;
```

```
popularSport = static_cast<sports>(popularSport + 1);
```

a) HOCKEY

b) FOOTBALL

c) BASEBALL

d) BASKETBALL

## Амалий машғулот 18. Стандарт кутубхона функциялари

**Ишдан мақсад:** C++ дастурлаш тилида стандарт кутубхона функциялари билан ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда бир нечта стандарт кутубхона функцияларидан фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** 8x8 кўринишидаги квадрат матрицани шахмат тахтаси деб ҳисоблаб, экрандан киритилган жойда турган фарзиннинг юриш йўллари чоп қилинсин.

Дастур коди:

dastur.cpp файли:

```
#include <iostream>
#include <cmath>
using namespace std;

int main(int argc, char* argv[])
{
    char sh[8][8];
    int x,y;
    cout << "Farzin turgan joyni kiriting (1..8): ";
    cin >> x >> y;
    x--;
    y--;
    for (int i=0;i<8;i++)
    for (int j=0;j<8;j++)
        sh[i][j]='_';
    for (int i=0;i<8;i++)
    {
        for (int j=0;j<8;j++)
```

```

        if (x==i && y==j)
            sh[i][j]='F';
        else if (x==i || y==j || abs(x-i) == abs(y-j))
            sh[i][j]='X';
    }
    cout << endl;
    for (int i=0;i<8;i++)
    {
        for (int j=0;j<8;j++)
            cout << sh[i][j] << " ";
        cout << endl << endl;
    }
    cout << endl;
    system("pause");
    return 0;
}

```

Дастур ишлаши натижаси:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
Farzin turgan joyni kiriting <1..8>: 3 5
- - X - X - X -
- - - X X X - -
X X X X F X X X
- - - X X X - -
- - X - X - X -
- X - - X - - X
X - - - X - - -
- - - - X - - -
Press any key to continue . . .

```

## Амалий топшириқлар

1. Қиймати 0 дан 20 гача бўлган  $k$  бутун сони берилган.  $k$ -тартибли Чебишев кўпхадли коеффитсентлари топилсин (Изоҳ: Чебишев кўпхадлари  $T_n(x)$  қуйидаги формула билан аниқланади:  
$$T_0(x) = 1, T_1(x) = x, T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), n = 2, 3, \dots$$
2. Ҳақиқий  $a_0, a_1, \dots, a_{15}$  сонлари берилган.  $(x-a_0)(x-a_1)\dots(x-a_{15})$  кўпхаднинг коеффисиентлари топилсин.
3. Берилган 10-даражали  $P(x)$  ва 6-даражали  $Q(x)$  кўпхаднинг коеффисиентлари бўйича  $P(Q(x))$  кўпхаднинг коеффисиентлари топилсин.
4. 10 та тош бўлиб, уларнинг оғирликлари мос равишда  $a_1, \dots, a_{10}$  бутун сонларга тенг. Берилган  $m_1, m_2, \dots, m_{10}$  оғирликлар учун  $c_1, \dots, c_{10}$  ҳосил қилинсин. Бу ерда,  $c_k - m_k$  оғирликни ҳосил қилиш усуллари сони, яни  $a_1x_1 + \dots + a_{10}x_{10} = m_k$  тенгламанинг ечимлари, бу ерда  $x_i (i = \overline{1,10})$  ўзгарувчиси 0 ёки 1 қиймат қабул қилади.
5. Сонлар ўқида  $n$  ( $n > 1$ ) та сон  $(a_1, a_2), (a_3, a_4), \dots, (a_{2n-1}, a_{2n})$  интерваллар жуфтлиги кўринишда берилган:
  - а) интервалларнинг умумий нуқталари борми?
  - б) Интерваллар бирлашмаси интервал ҳосил қилса, шу интервал учлари кўрсатилсин ( $L_1$  ва  $L_2$  интерваллар бирлашмаси деб шундай  $L_3$  интервалга айтиладики,  $L_1$  ва  $L_2$  интервалларга тегишли барча нуқталар  $L_3$  га ҳам тегишли бўлса).
  - д) интервалларнинг бирлашмасини  $n$  та кесишмайдиган интерваллар кўринишида тақдим этиш мумкин бўлган сон кўрсатилсин.
  - е) камида учта интервалга тегишли бутун сонлар бор бўлса, шу сонлардан бирортаси кўрсатилсин.
6. Текисликда  $n$  та ( $n \geq 4$ ) нуқта  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  координаталари билан берилган. Уларнинг ичида квадрат ҳосил қилувчи нуқталар бор ёки йўқлиги аниқлансин.
7. Маркази  $M_i (x_i, y_i)$  нуқтада бўлган  $r_i$  радиусли айланалар ( $i=1, \dots, n$ ) берилган. Қуйидагилар аниқлансин:
  - а) айланалар ичида учта кесишувчи айланалар борми?
  - б) алоҳида турган айланалар топилсин, яъни бошқа айланалар билан

умумий нуқталари йўқ, бирорта айлана ичида жойлашмаган ва бошқа айланаларни ўз ичига олмаган айланалар).

8.  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  координаталари билан берилган нуқталар тўпламининг медианалар сони топилсин ( $n > 2$  ва  $n$  жуфт сон). Тўпلام медианаси деб тўпلامнинг иккита ихтиёрий нуқтасини туташтирувчи шундай тўғри чизикқа айтиладики, унинг икки томонида бир ҳил сондаги нуқталар жойлашади ва бу тўғри чизикда ҳеч қандай учинчи нуқта ётмайди.
9. Арқон тортиш мусобақасида иштирок этмоқчи бўлган  $n$  та ўқувчилар ( $n$ -жуфт сон) икки гуруҳга бўлиниш учун айлана шаклида жойлашиб, бирдан то  $k$  сонигача санай бошладилар. Бунда ҳар  $k$ -ўқувчи даврадан чиқиб, иккинчи гуруҳга кўшилади. Санок даврада гуруҳларда ўқувчилар сони тенг бўлгунча давом этади. Ҳар бир ўқувчининг тартиб номери бошланғич даврада санок бошланган ўқувчидан бошланиб, санок йўналиши (соат миллари йўналиши) бўйича аниқланади. Берилган  $n$  ва  $k$  учун ҳар бир гуруҳдаги ўқувчилар тартиб номерлари аниқлансин.

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

### 1. ctype (ctype.h) кутубхона файлининг функциялари

isalnum(ch)	ch bu char turidagi o'zgaruvchi	<p>Функция куйидаги бутун турдаги кийматларни қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи киймати 'A'-'Z', 'a'-'z', '0'-'9' оралиғида бўлса 1 (true);</li> <li>• акс ҳолда 0 (false) киймат қайтаради;</li> </ul>
isctrl(ch)	ch bu char turidagi o'zgaruvchi	<p>Функция куйидаги бутун турдаги кийматларни қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи киймати ASCII жадвалидаги 0-31 ва 127 ) бўлса 1(true);</li> <li>• акс ҳолда 0 (false);</li> </ul>
isdigit(ch)	ch bu char turidagi o'zgaruvchi	<p>Функция куйидаги бутун турдаги кийматларни қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи киймати '0' - '9' белгилар белгиларини қабул қилса 1(true);</li> <li>• акс ҳолда, 0(false);</li> </ul>
islower(ch)	ch bu char turidagi o'zgaruvchi	<p>Функция куйидаги бутун турдаги кийматларни қайтаради:</p> <ul style="list-style-type: none"> <li>• агар ch ўзгарувчи киймати 'a' - 'z' оралиғидаги белгилардан бири бўлса, нол бўлмаган киймат (true) қайтаради</li> <li>• акс ҳолда, 0(false);</li> </ul>

2. cmath (math.h) кутубхона файлининг функциялари.

<b>Функция прототиби</b>	<b>Бажарадиган амали</b>
int abs(int i)	i сонни абсолют қийматини қайтаради
double ceil(double x)	Ҳақиқий x қийматини унга энг яқин катта бутун сонгача айлантиради ва уни ҳақиқий кўринишда қайтаради
double cos(double x)	x радианга тенг бўлган бурчакни косинусини қайтаради
double exp(double x)	$e^x$ қийматни қайтаради
double fabs(double x)	Ҳақиқий сонни абсолют қийматини қайтаради
double floor(double x)	Ҳақиқий x қийматни энг яқин кичик сонга айлантиради ва уни ҳақиқий сон кўринишида қайтаради
double fmod(double x, double y)	x сонини у сонига бўлиш натижасидаги қолдиқни қайтаради. % амалига ўхшаган, фақат ҳақиқий сон қайтаради
double log(double x)	x сонининг натурал логарифмини қайтаради

## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур квадрат тенглама илдизларини топиши керак. Дастур ишга тушганда айрим қийматларда жавоб чиқмасдан, хатолик ҳақида хабар берилмоқда.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 7, b = 8;
if (a == b++)
    a += a;
else if(a % 2 == 0)
    a += b * b;
else
    a += b; cout<<a;
```

- a) 16
- b) 15
- c) 56
- d) 1

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 3, b = 3;
if (pow(a++, b) == 65) a += b*4;
else a *= ++b;
cout<<a;
```



- a) 16
- b) 64
- c) 15
- d) 12

3. Ушбу функция қандай қиймат қайтаради?

```
{  
    int k = 0;  
    while(n > 0)  
    {  
        k++;  
        n /= 10;  
    }  
    return k;  
}
```

- a) n ниинг рақамлари сонини
- b) n нинг рақамлари йиғиндисини
- c) Ҳар доим нол қиймат қайтаради
- d) Қиймат қайтармайди

4. Ушбу функция қандай қиймат қайтаради?

```
int funksiya(int n)  
{ int s=0;  
    while(n > 0)  
    {  
        int a=n%10;  
        s=s+a;  
        n /= 10;  
    }  
    return s;
```

```
int funksiya(int n)  
{ int s=0;  
    while(n > 0)  
    {  
        int a=n%10;
```

```

    s=s+a;
    n /= 10;
}
return s;
}

```

- a) n ниинг рақамлари йиғиндисини
- b) n нинг рақамлари йиғиндисини
- c) Ҳар доим нол қиймат қайтаради
- d) n ниинг рақамлари сонини

5. Ушбу функция қандай қиймат қайтаради?

```

int funksiya(int n)
{
    int s=0;
    while(n > 0)
    {
        int a=n%10;
        s=s*10+a;
        n=n/10;
    }
    return s;
}

```

- a) n рақаминиинг тескарисини
- b) n нинг рақамлари йиғиндисини
- c) n ниинг рақамлари сонини
- d) Қиймат қайтармайди

6. Программа ишлаши натижасида экранга нима чиқади?

```

int F(int n, int k=15, int t=12)
{
    return n*k/t;
}
void main()
{
    int n=7, a[]={5,2,-3,2,4}, c=2, k=2;

```

```
for (int i=0;i<5;i++)
if (2*F(a[i])-F(n-a[i],c)>F(a[i], a[i], a[i/2])) c=c+F(a[i], n);
cout << c;
}
```

- a) 7
- b) 8
- c) 15
- d) 12

7. Программа ишлаши натижасида экранга нима чиқади?

```
int F(int n, int k=5)
{
k=3;
return n-k;
}
void main()
{
int n=7, a[]={5,2,3,2,4}, c=2, k=2;
for (int i=0;i<5;i++)
if (2*F(a[i])-F(n-a[i],c)>0) c=c+F(a[i], n);
cout << c;
}
```

- a) 5
- b) 6
- c) 7
- d) 8

8. Қуйидаги дастур кодининг натижасини топинг.

```
num2 = 4 / 5 - 4 % 5;
cout<< num2 <<endl;
```

- a) -4
- b) 4
- c) 5
- d) 11

9. Қуйидаги дастур кодининг натижасини топинг.

```
num2 = 4 * 3 + 7 / 5 - 25.5;
```

```
cout << num2 << endl;
```

a) -12.5

b) 10.5

c) -12

d) 12

10. Қуйидаги дастур кодининг натижасини топинг.

```
num = 6;
```

```
num = num + 2;
```

```
cout << num << endl;
```

a) 8

b) 10

c) -2

d) 2

## Амалий машғулот 19.

### Кўрсаткичлар ва адрес олувчи ўзгарувчилар

**Ишдан мақсад:** C++ дастурлаш тилида кўрсаткичлар ва адрес олувчи ўзгарувчилар билан ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда кўрсаткичлар ва адрес олувчи ўзгарувчилардан фойдалана олиш.

**Масаланинг қўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Икки соннинг ЭКУБини топувчи рекурсив функция тузилсин. Функцияга кўрсаткич орқали мурожаат қилинсин ва натижа экранга чоп қилинсин.

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;

int EKUB(int, int );
int main(int argc, char* argv[])
{
    int (*ekub_korsatkich)(int, int);
    ekub_korsatkich=EKUB;
    int a, b;
    cout << "Birinchi sonni kiriting: ";
    cin >> a;
    cout << "Ikkinchi sonni kiriting: ";
    cin >> b;
    cout << "EKUB = " << ekub_korsatkich(a, b) << endl;
    system("pause");
    return 0;
}

int EKUB(int son1, int son2)
```

```

{
if ( son2 == 0 )
    return son1;
return EKUB(son2, son1 % son2);
}

```

Дастур ишлаши натижаси:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
Birinchi sonni kiriting: 2431
Ikkinchi sonni kiriting: 2873
EKUB = 221
Press any key to continue . . .

```

### Амалий топшириқлар

1. Берилган  $n$  та ҳақиқий сонлар орасида қўшниларидан (ўзидан олдинги ва кейинги сонлардан) катта бўлган сонлар миқдори топилсин.
2. Берилган 10 та натурал сонларнинг энг катта умумий бўлувчиси топилсин.
3. Берилган  $n$  ва  $m$  натурал сонлари учун  $s = \sum_{i=1}^n \prod_{j=5}^m (i + j)$  ифоданинг қиймати ҳисоблансин.
4. Берилган  $n$  на натурал сонлардан иборат кетма-кетликнинг тартиб номерлари Фибоначчи сонлари бўлган ҳадларининг йиғиндиси ҳисоблансин.
5. 0 дан 15 гача бўлган сонлар иккилик кўринишида чоп қилинсин.

6. Берилган  $x$ ,  $y$ ,  $z$  векторларни элемент туридаги объектлар тўплами деб қараб (агарда  $k$  элемент  $x$  тўпламга тегишли бўлса  $x[k]=true$ , акс ҳолда  $x[k]=false$  ва ҳоказо), ушбу вектор - тўпламлар устида қуйидагилар амалга оширилсин:
- а) агар  $x$  тўплам  $y$  тўпламнинг қисм тўплами бўлса,  $y$  ҳолда  $t$  ўзгарувчига  $true$  қиймати, акс ҳолда  $false$  берилсин;
- б)  $z = x \cap y$  - тўпламлар кесишмаси топилсин;
- д)  $z = x \cup y$  - тўпламларнинг бирлашмаси топилсин;
- е)  $z = x \setminus y$  - тўпламларнинг айирмаси топилсин ( $z$  тўпламга  $x$  тўпламнинг  $y$  тўпламга кирмаган барча элементлари кирази).
7. `const int n = 20;`  
`float B[n][n];`  
 Қўшимча матрисадан фойдаланмаган ҳолда  $B$  матрицанинг транспонерланган кўриниши  $B^T$  ҳосил қилинсин.
8.  $n$  натурал сони ва 5-чи тартибли ҳақиқий турдаги квадрат матрица берилган. Бу матрицанинг  $n$ -даражаси топилсин ( $A^1=A$ ,  $A^2=AA$ ,  $A^3=A^2A$  ва ҳоказо).
9. `const int n=20;`  
`float nuqta[n][2], d;`  
 Матрицанинг сатр элементларини текисликдаги нуқталарнинг координаталари деб қараб, шу нуқталар орасидаги энг катта масофа топилсин.

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

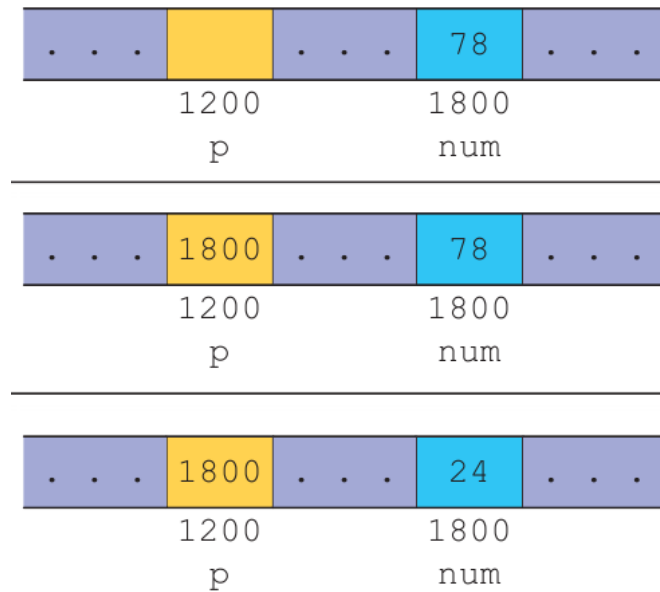
## Тарқатма материаллар.

1. Кўрсаткич ва адрес олиш амали ишлашига намуна

a. `num = 78;`

b. `p = &num;`

c. `*p = 24;`



## Динамик хотирадан фой ажратиш

2. Функцияга кўрсаткич эълони.

```
double f1(double x){return 5*sin(3*x)+x;}
```

```
double f3(double x){return x*x+1;}
```

```
double Integral(double(*f)(double),double a, double b)
```

## Кейслар банки

**Кейс 1.** Функцияга кўрсаткич орқали ишловчи дастур тузилди. Дастур вазифаси берилган интегрални эпсилон аниқликда ҳисоблаш. Лекин дастур ишлаши натижасида эпсилон аниқликда ҳисоблашга эришилмади.

### Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).



- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

### Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 16, b = 5;
if (sqrt(a) == b) a *= ++a;
else a *= ++b;
cout<<a;
```

- a) 96
- b) 15
- c) 56
- d) 19

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 3, b = 3;
if (pow(a++, b) == 65) a += b*4;
else a *= ++b;
cout<<a;
```

- a) 16
- b) 15
- c) 26
- d) 19

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 3, b = -1;
if (pow(b,a) == 1) a = a / (b++);
else a = b / (b--);
cout<<a;
```

- a) 1
- b) 5
- c) 2
- d) 9

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 0;
for (int i = 0; i < 50; i+=5)
{
    if (i / 10 >= 3)
        if (i % 10 != 5)
            a += i;
}
cout<<a;
```

- a) 70
- b) 75
- c) 85
- d) 80

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int i = 2, a = 0, b = 0;
while (i <= b)
{
    i = i + 2;
    a+=i%10;
    b = b + 3/2;
}
cout<<a;
```

- a) 0
- b) 3
- c) 2
- d) 1

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={ 15,30,25,10,45,20};
    int s=0;
for(int i=1;i<5;i++) if(a[i]>a[i-1]&&a[i]>a[i+1]) s+=a[i];
    cout<<s;
```

- a) 75
- b) 125
- c) 110
- d) 85

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={35,28,37,22,11,17};
    int max=a[0];
    for(int i=1;i<6;i++)
        if(a[i]>max) max=a[i];
    cout<<max;
```

- a) 37
- b) 35
- c) 11
- d) 17

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int &n, int k=3)
{
k=n-k;
return n-k;
}
void main()
{
int n=7, a[]={6,2,3,8,1}, c=5, k=2;
for (int i=0;i<5;i++)
if (F(n-c,c)) c=c+F(a[i], c);
cout << c;
}
```

- a) 150
- b) 250
- c) 160
- d) 180

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int a)
{
return a*9;
}
void main()
{
int a=2;
int t = a+F(a+1);
cout << t;
}
```

- a) 29
- b) 18
- c) 31
- d) 27

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=4;
int a[n][n], s=0;
for(int j=0;j<n;j++)
{ a[0][j]=1;
for(int i=1;i<n;i++) a[i][j]=2*a[i-1][j];
}
for(int i=0;i<n;i++) s+=a[i][i]+a[i][n-1-i];
cout<<s;
```

- a) 30
- b) 20
- c) 25
- d) 35

## Амалий машғулот 20-21.

### Динамик массивлар

**Ишдан мақсад:** C++ дастурлаш тилида динамик массивлар билан ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда бир ўлчамли ва кўп ўлчамли динамик массивлардан фойдалана олиш.

**Масаланинг қўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Берилган  $s$  векторнинг қийматлари иккиннинг даражаларига (1,2,4,8,16,...) тенг бўлган элементларнинг индекслари ва уларнинг йиғиндиси чоп қилинсин.

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;

int main(){

    int *s, n, k;
    cout << "Massiv elementlari soni = ";
    cin >> n;
    s = new int[n];
    for (int i = 0; i < n; i++)
    {
        cout << "s[" << i + 1 << "]=";
        cin >> s[i];
    }
    k = 0;
    cout << "Indekslar: ";
    for (int i = 0; i < n; i++)
    {
```

```

bool b = false;
unsigned long int d = 1;
for (;;)
{
    if (s[i] == d)
    {
        k = k + i;
        cout << i << " ";
        break;
    }
    d = d * 2;
}
cout << "Indekslar yig'indisi = " << k << endl;
system("pause");
return 0;
}

```

Дастур ишлаши натижаси:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
Massiv elementlari soni = 10
s[1]=4
s[2]=2
s[3]=7
s[4]=10
s[5]=64
s[6]=31
s[7]=28
s[8]=121
s[9]=128
s[10]=42
Indekslar: 0 1 4 8
Indekslar yig'indisi = 13
Press any key to continue . . .

```

## Амалий топшириқлар

1. Берилган  $s$  вектор катталиқнинг индекслари қуйидагиларга тенг бўлган элементлари чоп қилинсин:
  - а) иккиннинг даражалари: (1,2,4,8,16,...);
  - б) тўлиқ квадратлар: (1,4,9,16,25,...);
  - д) Фибоначчи сонлари: (1,2,3,5,8,13,...);
2. float \* x;

Берилган  $x$  вектор орқали  $x'$  вектор қуйидаги қоида бўйича ҳосил қилинсин ( $x'_k$  ҳосил бўлган векторнинг  $k$ -элементи қиймати):

  - а)  $x'_k = \max \{x_i\}$  бунда  $1 \leq i \leq k$ ;
  - б) вектор элементлари тескари тартибда жойлаштирилсин;
  - д)  $x'_1 = x_1, x'_n = x_n, x'_k = (x_{k-1} + x_k + x_{k+1})/3, k = 2, 3, \dots, n-1$ ;
  - е) вектор элементлари сиклик равишда  $n$  тага чапга сурилсин.
3. Берилган  $x$  векторнинг барча манфий элементлари унинг бошига ўтказилсин. Ҳосил бўлган векторда манфий ва қолган элементларининг бошланғич ўзаро жойлашуви сақлансин (қўшимча вектордан фойдаланилмасин).
4. Берилган  $x$  ва  $y$  векторларнинг ҳар бирида элементлар камаймайдиган тартибда жойлашган. Бу икки векторни бирлаштириб, шундай  $z$  вектор ҳосил қилинсинки, унинг элементлари ҳам камаймайдиган тартибда бўлсин.
5. Массив кўринишида берилган  $suz1$  ва  $suz2$  сўзларнинг ҳар бирида белгилар такрорланиб келмайди деб ҳисоблаган ҳолда, уларда қатнашаётган белгиларнинг жойлашуви билан фарқ қилса, тенг ўзгарувчига true, акс ҳолда false қиймат берилсин.
6. Берилган  $x$  вектор алмаштириш усули орқали камаймайдиган кўринишда тартиблансин (“нуфакчалар усули”). Бу усулнинг 1-қадамида қўшни  $x_k$  ва  $x_{k+1}$  ( $k=1, 2, 3, \dots, n-1$ ) элементлар кетма-кет солиштирилади ва агарда  $x_k > x_{k+1}$  бўлса, уларнинг ўрни алмаштирилади. Шу йўл билан энг катта элементни векторнинг охирида жойлашувига эришилади. Кейинги қадамда бу усул охириги элементдан ташқари барча элементларга қўлланилади ва ҳ.к. Жараённинг бирорта қадамида элементларни алмаштириш рўй бермаса тартибланиш тўхтатилсин.

7. Берилган  $x$  вектор орасига қўйиш усули орқали камаймайдиган кўринишда тартибланиш. Бу усулда векторнинг биринчи  $k$  та элементи камаймайдиган кўринишда тартибланган деб ҳисобланади;  $k+1$  - элементи олинади ва у биринчи  $k$  та элемент орасига шундай жойлаштириладики, ҳосил бўлган  $k+1$  та элемент тартибланган бўлади; бу усул  $k$  ўзгарувчининг 0 дан  $n-2$  гача қийматлари учун такрорланади.
8. Нуқта билан тугайдиган кичик лотин иборат матн берилган. Шу матнга фақат бир мартадан кирувчи барча ҳарфлар алфавит тартибида чоп қилинсин.
9. `float **A,**B,**C; // n x n ўлчамли матрицалар`  
`float *x,*y; // n ўлчамли вектор`  
 Берилган натурал  $n$  учун қуйидагилар ҳисоблансин:  
 а)  $C=A+B$ ; б)  $y=Ax$ ; д)  $C=AB$ .
10. Берилган бутун  $n$  ва  $m$  бўйича  $n \times m$  ўлчамли ҳақиқий турдаги динамик матрица берилган. Унинг сатрлари камаймайдиган кўринишда тартиблансин:  
 а) биринчи элементлар бўйича;  
 б) элементлар йиғиндиси бўйича;  
 д) энг катта элементлари бўйича.
11. Шахмат мусобақасида қатнашган  $n$  та шахматчининг натижалари  $T$  жадвалда берилган ( $n > 2$ ):  
`enum Uyin_Natijasi{ Y,D,M,X};`  
`Uyin_Natijasi` туридаги  $n \times n$  ўлчамли динамик массив (жадвал) ҳосил қилинсин. Унинг қийматлари қуйидагича аниқласин:  $jadval[i][j]=Y$ , агар  $i$ - иштирокчи  $j$ - иштирокчи устидан ғалаба қозонган бўлса (бунда  $jadval[j][i]=M$ ),  $jadval[i][j]=D$  ва  $jadval[j][i]=D$ , агар  $i$ - ва  $j$ - иштирокчилар дуранг ўйнаган бўлса, ҳамда  $jadval[i][i]=X$ .
- Жадвалнинг кўриниши қуйидагича бўлиши мумкин (расмга қаранг). Ютуқ учун 3 очко, дуранг учун 1 очко, мағлубият учун 0 очко берилади. Иштирокчилар номерлари уларнинг тўплаган очколари бўйича ўсмайдиган тартибда чоп қилинсин.

$$\begin{pmatrix} X & Y & M \\ M & X & D \\ Y & D & X \end{pmatrix}$$



12. Берилган  $n \times m$  ўлчамли бутун турдаги динамик равишда яратилган матритсанинг эгар нуқтаси индекслари чоп этилсин. Матритса элементи “*эгар нуқта*” дейилади, агарда у бир вақтнинг ўзида шу элемент жойлашган сатрдаги энг кичик ва устундаги энг катта элемент бўлса ёки аксинча.
13.  $n$  ўлчовли чизикли фазода  $m$  та вектор координаталари билан берилган. Бу векторлар чизикли эрки бўладими? Векторлар чизикли эрки дейилади, агар ҳар қандай  $i \neq j (1 \leq i, j \leq m)$  учун  $x_j = \alpha x_i$  шарти ўринли бўлмаса, бу ерда  $\alpha > 0$ .
14. Элементлари  $n$  та хақиқий сонлардан иборат  $x$ ,  $y$  ва  $z$  векторлар берилган.  $(a, a) - (b, c)$  катталиқ ҳисоблансин, бу ерда  $a$  вектор берилган векторлар ичидан энг катта минимал элементга эга (бундай вектор ягона деб ҳисоблансин),  $b$  ва  $c$  қолган иккита вектор,  $(p, q)$ -  $p$  ва  $q$  векторларнинг скаляр кўпайтмаси.
15.  $9 \times 4$  ўлчамли учта бутун турдаги матритсалар динамик равишда аниқланиб қийматлари берилган. Фақат ноллардан иборат сатрлари энг кўп бўлган матритса чоп этилсин (агар бундай матритсалар бир нечта бўлса, барчаси чоп этилсин).

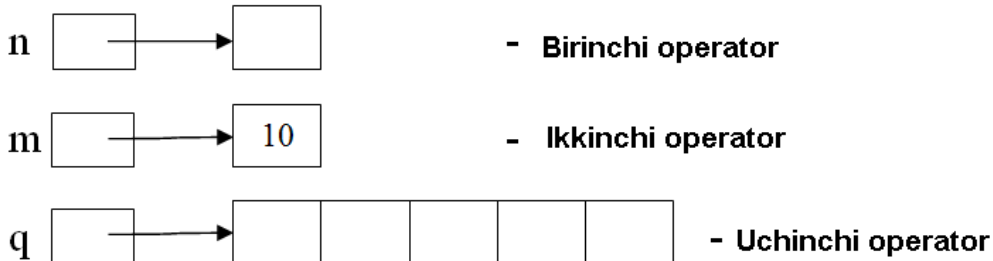
### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

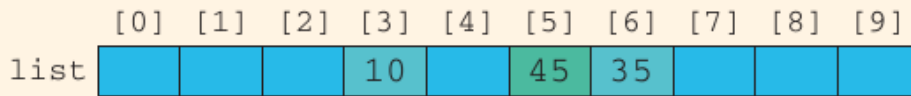
1.

```
int * n=new int;      // birinchi operator
int * m=new int(10);  // ikkinchi operator
int * q=new int[5];   // uchinchi operator
```



## Динамик хотирадан фой ажратиш

2. Массив хотирада кўриниши.



3. Динамик массив кўрсаткич орқали муружаат.



## Кейслар банки

**Кейс 1.** Динамик массивлар билан ишлайдиган дастур тузилди. Дастур вазифаси матрица қаторлари элементларини марказига нисбатан симметрик алмаштиришдан иборат. Дастур ишлаши натижасида матрица ўз ҳолатини ўзгартирмади.

## Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 7, b = 8;
if (a == b++)
    a += a;
else if(a % 2 == 0)
    a += b * b;
else
    a += b; cout<<a;
```

- a) 16
- b) 15
- c) 56
- d) 1

2. C++ тилида икки ўлчамли динамик массив учун хотирадан ажритилган жой тўғри кўрсатилган жавобни танланг.

- a) `int **a, n=10;`  
`a=new int *[n];`  
`for(int i=0;i<n;i++) a[i]=new int[n];`
- b) `int **a, n=10;`  
`a=new int **[n];`
- c) `int **a, n=10;`  
`a=new int [n];`  
`for(int i=0;i<n;i++) a[i]=new int[n];`
- d) `int **a, n=10;`  
`a=new int [n][n];`

3. C++ тилида \*\*a икки ўлчамли динамик массив учун хотирадан ажритилган жойни ўчириш тўғри кўрсатилган жавобни танланг.

a) for(i=0;i<m;i++) delete[]a[i];

delete[]a;

b) for(i=0;i<m;i++) delete[]a[i];

c) delete [[]];

d) for(i=0;i<m;i++) delete a[i];

delete[]a;

4. C++ тилида \*a бир ўлчамли динамик массив учун хотирадан ажритилган жойни ўчириш тўғри кўрсатилган жавобни танланг?

a) delete[]a;

b) delete [];

c) delete a;

d) delete [a];

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=3;
int a[n][n]={ {7,2,3},{3,2,4},{0,6,4} };
int s=0;
for(int i=0;i<n;i++)
    s+=a[0][i]+a[i][n-1];
cout<<s;
```

a) 23

b) 31

c) 21

d) 11

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={ 15,30,25,10,45,20};
int s=0;
for(int i=1;i<5;i++) if(a[i]>a[i-1]&& a[i]>a[i+1]) s+=a[i];
cout<<s;
```

- a) 75
- b) 125
- c) 110
- d) 85

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={35,28,37,22,11,17};  
    int max=a[0];  
    for(int i=1;i<6;i++)  
        if(a[i]>max) max=a[i];  
    cout<<max;
```

- a) 37
- b) 35
- c) 11
- d) 17

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=4;  
int a[n][n], s=0;  
for(int i=0;i<n;i++)  
    { a[i][0]=1;  
      for(int j=1;j<n;j++) a[i][j]=2*a[i][j-1];  
    }  
for(int i=0;i<n;i++) s+=a[i][i];  
cout<<s;
```

- a) 15
- b) 25
- c) 20
- d) 18

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=3;
int a[n][n]={ {7,2,8},{3,2,4},{4,6,9} };
int s=0;
for(int i=0;i<n;i++) s+=a[n/2][i];
cout<<s;
```

- a) 9
- b) 18
- c) 13
- d) 2

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=4;
int a[n][n], s=0;
for(int j=0;j<n;j++)
{
    a[0][j]=1;
    for(int i=1;i<n;i++) a[i][j]=2*a[i-1][j];
}
for(int i=0;i<n;i++) s+=a[i][i]+a[i][n-1-i];
cout<<s;
```

- a) 30
- b) 20
- c) 25
- d) 35

## Амалий машғулот 22-23.

### Сатрлар. Сатр устида амаллар. Сатр функциялари

**Ишдан мақсад:** C++ дастурлаш тилида сатрлар билан ишлаш, сатр устида турли амаллар бажариш, сатр функцияларидан фойдаланиш кўникмаларига эга бўлиш. Дастурда турли узунликдаги сатрлардан (ASCIIZ сатрлар, string туридаги сатрлар) фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** ASCIIZ сатр кўринишида берилган гапдаги симметрик сўзлар чоп этилсин. Гап нукта билан тугайди ва ундаги сўзлар бир-бирдан ‘,’ (вергул) ёки ‘ ’ (пробел) билан ажратилган.

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;
#include <cstring>
int main()
{
    const int n=80;
    char Satr[n];
    cout<<"Satrni kiriting:";
    cin.getline(Satr,sizeof(Satr));
    float s=0;
    int k;
    for(int i=0;i<strlen(Satr); i++)
        if(Satr[i]!=' ')
        {
            k=0;
            for(int j=0;j<strlen(Satr); j++)
                if(Satr[i]==Satr[j]||abs(Satr[i]-Satr[j])==32)
```

```

        k++;
        s+=1./k;
    }
    cout<<"Satrdagi turli harflar miqdori: "<<(int)s;
    system("pause");
    return 0;
}
Дастур ишлаши натижаси:

```

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
Satrni kiriting: Men UzMU talabasiman!
Satrdagi turli harflar miqdori: 12
Press any key to continue . . . _

```

### Амалий топшириқлар

1. char t[100];  
Берилган t сатрдаги бир-биридан фарқли бўлган кичик лотин ҳарфлари алфавит тартибида чоп қилинсин.
2. char c[80];  
Берилган с сатр қуйидагича ўзгартирилсин:
  - а) сатрдаги барча “bir” сатростилари “\_1\_” сатростига алмаштирилсин (бу ерда ‘\_’ - пробел);
  - б) сатрга кирувчи охириги ‘x’ ҳарфи (агарда у мавжуд бўлса) “ks” га алмаштирилсин;
  - д) сатрга кирувчи барча “th” белгилар ўчирилсин;
  - е) сатрга кирувчи ҳар бир ‘q’ ҳарфидан кейин ‘u’ ҳарфи қўшилсин.



3. `char c1[10],c2[10],c3[11];`  
Ўнлик саноқ системасидаги бутун `c1`, `c2` сонларнинг разряд бўйича йиғиндиси `c3` ўзгарувчисида ҳосил қилинсин.
4. `char bayt[8];`  
Берилган бутун `n` ( $0 \leq n \leq 127$ ) сонига қарама-қарши соннинг (манфий соннинг) қўшимча коддаги иккилик кўриниши байт массивида ҳосил қилинсин.
5. `char gap[80];`  
Берилган гапдаги сўзлар бир-биридан ‘,’ ёки ‘\_’ (пробел) белгиси билан ажратилган ва гап нукта билан тугайди (гап ўзгарувчисининг қиймати). Қуйидагилар чоп қилинсин:
  - а) кетма-кетликда фақат бир марта учраган сўзлар;
  - б) кетма-кетликда бир неча бор учраган сўзлар;
  - д) кетма-кетликдаги барча сўзлар алфавит тартибида.
6. `char gap[80];`  
Берилган гапдаги сўзлар бир-биридан ‘,’ ёки ‘\_’ (пробел) белгиси билан ажратилган ва гап нукта билан тугайди (гап ўзгарувчисининг қиймати). Қуйидаги шартларни бажарувчи сўзлар чоп қилинсин:
  - а) биринчи ҳарфи яна учраган;
  - б) энг узун;
  - д) ҳарфлари такрорланмайдиган.
7. Ҳар бири 2 тадан 10 тагача латин ҳарфларидан иборат 2 дан 30 гача бўлган сўзлар кетма-кетлиги берилган. Сўзлар бир-бири билан камида битта пробел билан ажратилган ва охириги сўз нукта билан тугайди. Кетма-кетликнинг охириги сўзидан фарқли барча сўзлар қуйидаги қоида бўйича ўзгартирилсин:
  - а) сўздаги биринчи ҳарфи унинг охирига ўтказилсин;
  - б) сўздаги охириги ҳарф унинг бошига ўтказилсин;
  - д) сўздаги биринчи ҳарф олиб ташлансин.
8. Ҳар бири 2 тадан 10 тагача латин ҳарфларидан иборат 2 дан 30 гача бўлган сўзлар кетма-кетлиги берилган. Сўзлар бир-бири билан камида битта пробел билан ажратилган ва охириги сўз нукта билан тугайди. Кетма-кетликнинг охириги сўзидан фарқли барча сўзлари қуйидаги қоида бўйича ўзгартирилсин:

- а) сўздаги охирги ҳарф такрорланса, охиргисидан бошқа такрорланганлари олиб ташлансин;
- б) сўздаги ҳар бир ҳарфни фақат биринчи марта қатнашгани қолдирилиб, қолганлари олиб ташлансин;
- д) агар сўзнинг узунлиги тоқ сонга тенг бўлса, у ҳолда унинг ўртасидаги ҳарф олиб ташлансин.
9. 1 дан 1999 гача бўлган бутун н сонининг Рим рақамларидаги кўриниши чоп қилинсин.
10. Рақам ва тўртта арифметик амаллардан ('\*', '+', '-', '\*') ташкил топган ифода постфикс кўринишига ўтказилсин. Постфикс шаклда олдин операндлар ёзилади, кейин амаллар.

Мисоллар:	оддий ёзув	постфикс ёзув
	3+4	34+
	(5-4)+2	54-2+
	2*(3+4)*5	234+*5*

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

### 1. string туридаги сатрлар устидан амаллар

Амал	Мазмуни	Мисол
=, +=	Қиймат бериш амали	s="satr01234" s+="2satr000"
+	Сатрлар улаш амали (конкантенация)	s1+s2
==, !=, <, <=, >, >=	Сатрларни солиштириш амаллари	s1==s2    s1>s2 && s1!=s2
[]	Индекс бериш	s[4]
<<	Оқимга чиқариш	cout << s
>>	Оқимдан ўқиш	cin >> s (probelgacha)

### 2. C++ тилидаги белги ўзгармаслар.

Белгилар синфлари	Белги ўзгармаслар
Катта ҳарфлар	'A' ... 'Z', 'A' ... 'YA'
Кичик ҳарфлар	'a' ... 'z', 'a' ... 'ya'
Рақамлар	'0' ... '9'
Бўш жой	gorizontal tabulyasiya (ASCII kodi 9), satrni o'tkazish (ASCII kodi 10), vertikal tabulyasiya (ASCII kodi 11), formani o'tkazish (ASCII kodi 12), karetkani qaytarish (ASCII kodi 13)
Пунктуатсия белгилари (ажратувчилар)	! " # \$ % & ' ( ) * + - , . / : ; < = > ? @ [ \ ] ^ _ {   } ~
Бошқарув белгилари	ASCII kodi 0...1Fh oraliq'ida va 7Fh bo'lgan belgilar
Пробел	ASCII kodi 32 bo'lgan belgi
Ўн олтилик рақамлар	'0' ... '9', 'A' ... 'F', 'a' ... 'f'

## Кейслар банки

**Кейс 1.** Сатрлар билан ишлайдиган дастур тузилди. Дастур вазифаси сатрда ҳарфлар нечтадан қатнашганини аниқлашдан иборат. Дастур ишлаши натижасида қайтарилган ҳарфларни ҳар бири учун миқдорини алоҳида ҳисобламоқда. Аслида қайтарилган ҳарфларни бир марта ҳисоблаши керак. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="ABBA", s2[]="BC", s3[]="CCBA";
```

```
strncat(s2,s3,2);
```

```
strcpy(s1,s2);
```

```
cout << s2;
```

a) ABC

b) BCCC

c) CBVV

d) BCCA

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="BAB", s2[]="ABC", s3[]="CBA";  
string a="ABC";  
if (s1[0]==s1[1] && s2[0])  
cout<<s3;  
else cout << a[1];
```

- a) B
- b) C
- c) A
- d) AB

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
string s1="AB", s2="BCC", s3="CAC";  
int i=1;  
s1.insert(i,s3);  
s2.append(s3);  
s2.erase(1,2);  
s3.erase(2,1);  
s3=s1+s2.substr(1,1);  
cout<<s3;
```

- a) ACACBC
- b) ABACAB
- c) BCBAVC
- d) ABCABC

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="BAB", s2[]="ABC", s3[]="CBA";  
if (s1[0]==s1[1] && s3[0])  
cout<<s3;  
else cout << s2;
```

- a) ABC
- b) CBA
- c) BAC
- d) ACB

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char c[]={'a','i','o','e','\0'};
char matn[] = "Programmash asoslari va kompyuter texnologiyalari";
int k=0;
for(int i=0; i<strlen(c);i++){
    k=0;
    for(int j=0;j<strlen(matn);j++)
        if(matn[j]==c[i])
            k++;
    cout<<k<<"\t";
}
```

- a) 8 3 5 2
- b) 2 3 5 8
- c) 0 3 5 8
- d) a i o e

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="BAB", s2[]="ABC", s3[]="ABA";
string a="lalala";
if (strev(s3)==s3)
    cout<<strev(strcpy(s2,strev(s1)));
else cout << a.find("al");
```

- a) BAB
- b) ABA
- c) ABC
- d) CBA

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int s=0, m=0 ;
char matn[] = "Programmash asoslari fani 2015 - yil";
for(int i=0;i<strlen(matn);i++)
    if(isdigit(matn[i])) s++;
```

```
else if (isalpha(matn[i])) m++;  
cout<<s<<"\t"<<m;
```

- a) 4 28
- b) 2 31
- c) 5 29
- d) 6 23

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="ABBA", s2[]="BABC", s3[]="CCBA";  
strncpy(s2,s3,2);  
strncpy(s1,s2,1);  
cout << s1;
```

- a) CBBA
- b) ABBC
- c) ACCB
- d) BAAC

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
string s1="A", s2="B", s3="C";  
int i=1;  
s1.insert(i,s3);  
s1.append(s2);  
s2.insert(i,s3);  
s1.erase(1,2);  
s3=s1+s2;  
cout<<s3;
```

- a) ABC
- b) CAB
- c) ACB
- d) BBC

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char matn[] = "d2as#3!";  
for(int i=0;i<strlen(matn)-2;i++)  
    if(!isdigit(matn[i]) && isalpha(matn[i]))  
        cout<<matn[i];
```

- a) das
- b) asd
- c) 2#3
- d) 23



## **II СЕМЕСТР**

## IV. МАЪРУЗА МАТЕРИАЛЛАРИ

### Маъруза 22. Функция ва массивлар

#### Режа:

1. Бир ўлчамли массив функция параметри сифатида
2. Кўп ўлчамли массив функция параметри сифатида
3. Ўзгарувчан параметрли функциялар

Таянч иборалар: *include*, *Функция*, *Параметр*, *аргумент*, *Динамик*, *new*, *delete*, *Кўрсаткич*, *cout*, *cin*, *Ўзгарувчи*, *Идентификатор*

#### 22.1. Бир ўлчамли массив функция параметри сифатида

Функциялар массивни параметр сифатида ишлатиши ва уни функциянинг натижаси сифатида қайтариши мумкин. Агар массив параметр орқали функцияга узатилса, элементлар сонини аниқлаш муаммоси туғилади, чунки массив номидан унинг узунлигини аниқлашнинг иложи йўқ. Айрим ҳолларда, масалан, белгилар массиви сифатида аниқланган сатр (ASCIIZ сатрлар) билан ишлаганда массив узунлигини аниқлаш мумкин, чунки сатрлар ‘\0’ белгиси билан тугайди.

Мисол учун:

```
#include <iostream>
using namespace std;
int len(char s[]) //массивни параметр сифатида ишлатиш
{
    int m=0;
    while(s[m++]);
    return m-1;
}
```

```

}
void main ()
{
char z[]="Ushbu satr uzunligi = ";
cout << z << len(z);
}

```

Функция параметри сифатида сонлар массиви жўнатилганда массив элементлари сони алоҳида жўнатилади. Массивнинг ўзи ўлчамисиз ёзилган ҳолда функцияда эълон қилинади.

```

int sumArray(const int list[], int listSize)
{
int index;
int sum = 0;
for (index = 0; index < listSize; index++)
sum = sum + list[index];
return sum;
}

```

Функция параметри сатр бўлмаган ҳолларда фиксирланган узунликдаги массивлар ишлатилади. Агар турли узунликдаги массивларни узатиш зарур бўлса, массив ўлчамларини параметр сифатида узатиш мумкин ёки бу мақсадда глобал ўзгарувчидан фойдаланишга тўғри келади.

```

#include <iostream>
using namespace std;
float sum(int n,float *x)
{
float s=0;
for (int i=0;i<n;i++)s+=x[i];
return s;
}

```

```

}
void main()
{
    float E[]={1.2,2.0,3.0,4.5,-4.0};
    cout<<sum(5,E);
}

```

Массив номи кўрсаткич бўлганлиги сабабли массив элементларини функцияда ўзгартириш мумкин ва бу ўзгартиришлар функциядан чиққандан кейин ҳам сақланиб қолади.

```

#include <iostream>
using namespace std;
void vector_01(int n,int*x,int * y) //бу иккинчи усул
{
    for (int i=0;i<n;i++)
        y[i]=x[i]>0?1:0;
}
void main()
{
    int a[]={1,2,-4,3,-5,0,4};
    int c[7];
    vector_01(7,a,c);
    for(int i=0;i<7;i++) cout<<'\\t'<<c[i];
}

```

Масала. Бутун турдаги ва элементлари камаймайдиган ҳолда тартибланган бир ўлчамли иккита массивларни ягона массивга, тартибланиш сақланган ҳолда бирлаштириш амалга оширилсин.

```

#include <iostream>
using namespace std;

```

```

\\бутун турдаги массивга кўрсаткич қайтарадиган
\\функция
int * massiv_ulash(int,int*,int,int*);
void main()
{
    int c[]={-1,2,5,10},d[]={1,7,8};
    int * h;
    h=massiv_ulash(5,c,3,d);
    for(int i=0;i<8;i++) cout<<'\\t'<<h[i];
    delete[]h;
}

    int * massiv_ulash(int n,int *a ,int m,int *b);
{
    int * x=new int[n+m];
    int ia=0,ib=0,ix=0;
    while (ia<n && ib<m)
        a[ia]>b[ib]?x[ix++]=b[ib++]:x[ix++]=a[ia++];
    while(ib<m)x[ix++]=b[ib++];
    while(ia<n)x[ix++]=a[ia++];
return x;
}

```

## 22.2. Кўп ўлчамли массив функция параметри сифатида

Кўп ўлчамли массивлар билан ишлаш маълум бир мураккабликка эга, чунки массивлар хотирада жойлаш тартиби турли вариантда бўлиши мумкин. Масалан, функция параметрлар рўйхатида  $n \times n$  ўлчамдаги хақиқий турдаги  $x[n][n]$  массивга мос келувчи параметрни

```
float sum(float x[n][n])
```

кўринишда ёзиб бўлмайди. Муаммо ечими - бу массив ўлчамини параметр сифатида узатиш ва функция сарлавҳасини қуйидагича ёзиш керак:

```
float sum(int n,float x[][]);
```

Кўп ўлчамли массивларни параметр сифатида ишлатишда бир нечта усуллардан фойдаланиш мумкин.

1-усул. Массивнинг иккинчи ўлчамини ўзгармас ифода ( сон) билан кўрсатиш:

```
float sum(int n,float x[][10])
{float s=0.0;
  for(int i=0;i<n;i++)
    for(int j=0;j<n;j++)
      s+=x[i][j];
  return s;}
```

2-усул. Икки ўлчамли массив кўрсаткичлар массиви кўринишида аниқланган ҳолатлар учун кўрсаткичлар массивини (матрица сатрлар адресларини) бериш орқали:

```
float sum(int n,float *p[])
{
  float s=0.0;
  for(int i=0;i<n;i++)
    for(int j=0;j<n;j++)
      s+=p[i][j];//”*p[i][j]” эмас, чунки массивга мурожат
  return s;
}
void main()
{
```

```

float x[][4]={{11, -12, 13, 14},{21, 22, 23, 24},
             {31, 32, 33, 34},{41, 42, 43, 44}};
float *ptr[4];
for(int i=0;i<4;i++) ptr[i]=(float *)&x[i];
cout<<sum(4,ptr)<<endl;
}

```

3-усул. Кўрсаткичларга кўрсаткич кўринишида аниқланган динамик массивларни ишлатиш билан:

```

float sum(int n,float **x)
{
    float s=0.0;
    for(int i=0;i<n;i++)for(int j=0;j<n;j++)s+=x[i][j];
    return s;
}

void main()
{
    float **ptr;
    int n;
    cin>>n;
    ptr=new float *[n];
    for(int i=0;i<n;i++)
    {
        ptr[i]=new float [n];
        for(int j=0;j<n;j++)
            ptr[i][j]=(float)((i+1)*10+j);
    }
    cout<<sum(n,ptr);
}

```

```

    for(int i=0; i<n;i++) delete ptr[i];
    delete[]ptr;
}

```

Мисол: Берилган қийматни массив элементларидан қидириш функциясини қуринг:

```

#include <iostream>
using namespace std;
const int ARRAY_SIZE = 10;
int seqSearch(const int list[], int listLength,int
searchItem);
int main()
{
int intList[ARRAY_SIZE];
int number;
cout << " Enter " << ARRAY_SIZE << " integers." << endl;
for (int index = 0; index < ARRAY_SIZE; index++)
cin >> intList[index];
cout << endl;
cout << "Enter the number to be "<< "searched:";
cin >> number;
cout << endl;
int pos = seqSearch(intList, ARRAY_SIZE, number);
if (pos!= -1)
    cout << number<< " is found at position " << pos
    << endl;
else
cout << number<< " is not in the list." << endl;

```



```

    return 0;
}
int seqSearch(const int list[ ], int listLength, int
searchItem)
{
    int loc;
    bool found = false;
    loc = 0;
    while (loc < listLength && !found)
        if (list[loc]== searchItem) found = true;
            else loc++;
    if (found)
        return loc;
        else
    return -1;
}

```

Дастурни ишга тушириб куйидаги кийматларни киритамиз

Enter 10 integers.

2 56 34 25 73 46 89 10 5 16

Enter the number to be searched: 25

25 is found at position 3

### 22.3. Ўзгарувчан параметрли функциялар

C++ тилида параметрлар сони номаълум бўлган функцияларни ҳам ишлатиш мумкин. Бундан ташқари уларнинг турлари ҳам номаълум бўлиши мумкин. Параметрлар сони ва тури функцияни чақириш-даги аргументлар сони ва уларнинг турига қараб аниқланади. Бундай функциялар сарлавҳаси куйидаги форматда ёзилади:

<функция тури> <функция номи> (<ошкор параметрлар рўйхати>, ...)

Бу ерда <ошкор параметрлар рўйхати> - ошкор равишда ёзилган параметрлар номи ва тури. Бу параметрлар мажбурий параметрлар дейилади. Бундай параметрлардан камида биттаси бўлиши шарт. Қолган параметрлар сони ва тури номаълум ҳисобланади. Уларни аниқлаш ва ишлатиш тўла равишда дастур тузувчи зиммасига юкланади.

Ўзгарувчан сондаги параметрларни ташкил қилиш усули умуман олганда иккита:

1-усул. Параметрлар рўйхати охирида яна бир махсус параметр ёзилади ва унинг қиймати параметрлар тугаганлигини билдиради. Компилятор томонидан функция танасида параметрлар бирма-бир аниқлаштирилади. Барча параметрлар тури охирги махсус параметр тури билан устма-уст тушади деб ҳисобланади;

2-усул. Бирорта махсус параметр сифатида номаълум параметрлар сони киритилади ва унга қараб параметрлар сони аниқланади.

Иккала усулда ҳам параметрларга мурожаат қилиш учун кўрсаткичлар ишлатилади. Мисоллар келтирамиз.

1 - усул:

```
#include <iostream>
using namespace std;
float Sonlar_kupaytmasi(float arg,...)
{
    float p=1.0;
    float *ptr=&arg;
    if(*prt==0.0) return 0.0;
    for(;*prt;prt++)p*=*prt;
    return p;
}
```

```

void main()
{
cout<<Sonlar_kupaytmasi(2e0,3e0,4e0,0e0)<<' \n';
cout<<Sonlar_kupaytmasi(1.0,2.0,3.0,10.0,8.0,0.0);
}

```

Натижа:

24

480

2 - усул:

```

#include <iostream>
using namespace std;
int Yigindi(int,...);
void main()
{
cout<<"\nYigindi(2,6,4)="<<Yigindi(2,6,4);
cout<<"\nYigindi(6,1,2,3,4,5,6)="
cout<<Yigindi(6,1,2,3,4,5,6);
}
int Yigindi(int k,...)
{
int *ptr=&k
int s=0;
for(;k;k--)s+=*(++ptr);
return s;
}

```

Натижа:

Yigindi(2,6,4)=10

Yigindi(6,1,2,3,4,5,6)=21

Иккала мисолда ҳам номаълум параметрлар берилган махсус параметр турини қабул қилган. Ҳар хил турдаги параметрларни ишлатиш учун турни аниқлайдиган параметр киритиш керак:

```
#include <iostream>
using namespace std;
float Summa(char,int,...);
void main()
{
    cout<<Summa('i',3,10,20,30);
    cout<<Summa('f',3,10.0,20.0,5.0);
    cout<<Summa('d',3,10,20,30);
}
int Summa(char z,int k,...)
{
    switch(z)
    {
        case 'i':
        {
            int *ptr=&k+1; int s=0;
            for (;k--;ptr++) s+=*(ptr);
            return (float)s;
        }
        case 'f':
        {
            float*ptr=(float *)(&k+1); float s=0.0;
            for (;k--;ptr++) s+=*(ptr);
```

```

    return s;
}
default:
{
    cout<<"\n parametr hato berilgan";
    return 9999999.0;
}
}
}

```

Юқорида келтирилган мисолда номаълум параметрларни турини аниқлаш масаласи компилятор томонидан эмас, балки дастур тузувчиси томонидан ҳал қилинган.

#### **Адабиётлар рўйхати:**

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

#### **Мустақил таълим мавзулари:**

1. Массивлар функция қайтарувчи қиймат тури сифатида.
2. Функцияларда статик массивларни ишлатиш.
3. main() функциясига қийматлар жўнатиш.
4. Массивларни typedef орқали функцияга жўнатиш.

## Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include	include	препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
Функция	Function	программа алоҳида бўлаги, асосий қисм томонидан чақириб ишлатилади
Параметр	Parametr	функция ишлаши учун керак берилганлар
аргумент	Argument	функцияга параметрига жўнатиладиган қиймат
Динамик	Dinamic	қиймати дастур ишўлаши давомида ўзгариб турадиган берилган
new	New	хотирадан янги жой ажратиш
delete	Delete	хотирадан ажратилган жойни тозалаш
Кўрсаткич	pointer	қийматлари адрес бўлган ўзгарувчилар
cout	cout	экранга чиқариш оқими.
cin	cin	экрандан киритиш оқими.
Ўзгарувчи	Variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги.
Идентификатор	identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизик (‘_’) белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги.

## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастурда массивдаги туб сонлар миқдорини аниқловчи функция мавжуд. Дастур ишлаши натижасида туб сонлар миқдорини нотўғри аниқлади. Дастурни тўғри ишлашини таъминлаш керак.

**Кейсни бажариш босқчилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

### Назорат саволлари:

- 11.Массив тушунчасига таъриф беринг.
- 12.Қандай турдаги массивларни биласиз?
- 13.Массивлардан қандай мақсадларда фойдаланиш мумкин?
- 14.Сонлар массиви функция параметри сифатида қандай жўнатилади?
- 15.Бир ўлчамли массив функция параметри сифатида ишлатишга мисол келтиринг.
- 16.Кўп ўлчамли массивларни параметр сифатида ишлатишда қандай усуллардан фойдаланиш мумкин?
- 17.Берилган қийматни массив элементларидан қидириш функциясини куринг.
- 18.Ўзгарувчан параметрли функциялар сарлавхаси қандай ёзилади?
- 19.Ўзгарувчан сондаги параметрларни ташкил қилишнинг қандай усуллари мавжуд ва улар қай тарзда амалга оширилади?
- 20.Матрица киритиш ва чиқариш функцияларини куринг.

### Тест саволлари:

1. Қуйидаги дастур қисми ишлаши натижасида экранга нима чоп этилади?

```
int f(int a[],int n)
{
int m=a[0];
for(int i=1;i<n;i++)
    if(a[i]>m) m=a[i];
return m;
}
int main()
{
int a[]={23,23,45,12,34};
int n=5;
cout<<f(a,n);
}
```

- a) 45
- b) 23
- c) 12
- d) 34

2. Қуйидаги дастур қисми ишлаши натижасида экранга нима чоп этилади?

```
int f(int a[],int n)
{
    int m=a[0];
    for(int i=1;i<n;i++)
        if(a[i]<m) m=a[i];
return m;
}
int main()
{int a[]={23,23,45,12,34};
    int n=5;
cout<<f(a,n);
}
```

- a) 12
- b) 23
- c) 34
- d) 45

3. Қуйидаги дастур қисми ишлаши натижасида экранга нима чоп этилади?

```
int f(int a[],int n)
{    int m=0;
    for(int i=0;i<n;i++) m+=a[i];
return m; }
int main()
    int a[]={0,1,2,3,4};
    int n=5;
cout<<f(a,n); }
```

- a) 10
- b) 0
- c) 6
- d) 4



4. Қуйидаги дастур қисми ишлаши натижасида экранга нима чоп этилади?

```
int f(int a[],int n)
{   int m=0;
    for(int i=0;i<n;i++)
    if(a[i]%2) m+=a[i];
    return m; }
int main(){
    int a[]={0,1,2,3,4};
    int n=5;
    cout<<f(a,n); }
```

- a) 4
- b) 10
- c) 6
- d) 0

5. Қуйидаги дастур қисми ишлаши натижасида экранга нима чоп этилади?

```
int len(char s[])
{   int m=0;
    while(s[m++]);
    return m-1; }
int main()
{char z[]="Ushbu satr uzunligi = ";
    cout<<z<<len(z); }
```

- a) Ushbu satr uzunligi =22
- b) 22
- c) Ushbu satr uzunligi
- d) 0

6. Қуйидаги дастур қисми ишлаши натижасида экранга нима чоп этилади?

```
float sum(int n,float *x)
{ float s=0;
    for (int i=0;i<n;i++)s+=x[i];
    return s; }
void main()
{ float E[]={1.2,2.0,3.0,4.5,-4.0};
    cout<<sum(5,E); }
```

- a) 6.7
- b) 1.2
- c) 6.2
- d) 0

7. Куйидаги дастур қисми ишлаши натижасида экранга нима чоп этилади?

```
void vector(int n,int x[],int y[])
{ for (int i=0;i<n;i++)
  y[i]=x[i]>0?1:0;}
void main()
{ int a[]={1,2,-4,3,-5,0,4};
  int c[7];
  vector(7,a,c);
  for(int i=0;i<7;i++) cout<<' '<<c[i];}
```

- a) 1 1 0 1
- b) 1 1 1 0
- c) 1 0 0 0
- d) 1 1 1 1

8. Куйидаги дастур қисми ишлаши натижасида экранга нима чоп этилади?

```
void vector(int n,int x[],int y[])
{ for (int i=0;i<n;i++)
  y[i]=x[i]>0?1:0; }
void main()
{ int a[]={1,2,-4,3,-5,0,4};
  int c[7];
  vector(7,a,c);
  int s=0;
  for(int i=0;i<7;i++) s+=c[i];
  cout<<s;}
```

- a) 4
- b) 6.7
- c) 0
- d) 1

9. Қуйидаги дастур қисми ишлаши натижасида экранга нима чоп этилади?

```
void vector(int n,int x[])
{ int s=0;
  for (int i=0;i<n;i++)  s+=x[i]>0?1:0;
  cout<<s;
}
void main()
{
  int a[]={1,2,-4,3,-5,0,4};
vector(7,a);}
```

- a) 4
- b) 6.7
- c) 0
- d) 1

10. Қуйидаги дастур қисми ишлаши натижасида экранга нима чоп этилади?

```
void vector(int n,int x[])
{ int s=0;
  for (int i=0;i<n;i++)  if(x[i]<0) s+=x[i];
  cout<<s;
}
void main()
{
  int a[]={1,2,-4,3,-5,0,4};
vector(7,a);}
```

- a) -9
- b) 4
- c) 10
- d) 11

## Маъруза 23-24. Тузилмалар. Бирлашмалар

### Режа:

1. Структуралар.
2. Структура функция аргументи сифатида
3. Структуралар массиви
4. Структураларга кўрсаткич
5. Бирлашмалар ва улар устида амаллар

Таянч атамалар: *struct, union, cin, if, for, return, ўзгарувчи, идентификатор, new, unsigned, кўрсаткич.*

### 23.1. Структуралар

Маълумки, бирор предмет соҳасидаги масалани ечишда ундаги объектлар бир нечта, ҳар хил турдаги параметрлар билан аниқланиши мумкин. Масалан, текисликдаги нукта ҳақиқий турдаги  $x$ -абсисса ва  $y$ -ордината жуфтлиги -  $(x,y)$  кўринишида берилади. Талаба ҳақидаги маълумотлар: сатр туридаги талаба фамилия, исми ва шарифи, мутахассислик йўналиш, талаба яшаш адреси, бутун турдаги туғилган йили, ўқув босқичи, ҳақиқий турдаги рейтинг бали, мантиқий турдаги талаба жинси ҳақидаги маълумот ва бошқалардан шаклланади.

Дастурда ҳолат ёки тушунчани тавсифловчи ҳар бир берилганлар учун алоҳида ўзгарувчи аниқлаб масалани ечиш мумкин. Лекин бу ҳолда объект ҳақидаги маълумотлар «тарқоқ» бўлади, уларни қайта ишлаш мураккаблашади, объект ҳақидаги берилганларни яхлит ҳолда кўриш қийинлашади.

C++ тилида бир ёки ҳар хил турдаги берилганларни жамланмаси *структура* деб номланади. Структура фойдаланувчи томонидан аниқланган берилганларнинг янги тури ҳисобланади. Структура қуйидагича аниқланади:

```
struct <struktura nomi>
{
    <tur1 > <nom1>;
    <tur2 > <nom2>;
    ...
}
```

```
<turn> <nomn>;  
};
```

Бу ерда <структура номи> - структура кўринишида яратилаётган янги турнинг номи, “<tur<sub>i</sub>> <nom<sub>i</sub>>,” - структуранинг i-майдонининг (nom<sub>i</sub>) эълони.

Бошқача айтганда, структура эълон қилинган ўзгарувчилардан (майдонлардан) ташкил топади. Унга ҳар хил турдаги берилганларни ўз ичига олувчи *қобик* деб қараш мумкин. Қобикдаги берилганларни яхлит ҳолда кўчириш, ташқи курилмалар (бинар файлларга) ёзиш, ўқиш мумкин бўлади.

Талаба ҳақидаги берилганларни ўз ичига олувчи структура турининг эълон қилинишини кўрайлик.

```
struct Talaba  
{  
    char FISH[30];  
    unsigned int Tug_yil;  
    unsigned int Kurs;  
    char Yunalish[50];  
    float Reyting;  
    unsigned char Jinsi[5];  
    char Manzil[50];  
    bool status;  
};
```

Дастурда структуралардан фойдаланиш, шу турдаги ўзгарувчилар эълон қилиш ва уларни қайта ишлаш орқали амалга оширилади:

```
Talaba talaba;
```

Структура турини эълонида турнинг номи бўлмаслиги мумкин, лекин бу ҳолда структура аниқланишидан кейин албатта ўзгарувчилар номлари ёзилиши керак:

```
struct  
{  
    unsigned int x,y;  
    unsigned char Rang;  
} Nuqta1, Nuqta2;
```

Келтирилган мисолда структура туридаги Nuqta1, Nuqta2

Ўзгарувчилари эълон қилинган.

Структура туридаги ўзгарувчилар билан ишлаш, унинг майдонлари билан ишлашни англатади. Структура майдонига мурожаат қилиш ‘.’ (нуқта) орқали амалга оширилади. Бунда структура туридаги ўзгарувчи номи, ундан кейин нуқта қўйилади ва майдон ўзгарувчисининг номи ёзилади. Масалан, талаба ҳақидаги структура майдонларига мурожаат қуйидагича бўлади:

```
talaba.Kurs=2;
talaba.Tug_yil=1988;
strcpy(talaba.FISH, "Abdullaev A.A.");
strcpy(talaba.Yunalish,
"Informatika va Axborot texnologiyalari");
strcpy(talaba.Jinsi, "Erk");
strcpy(talaba.Manzil,
"Toshkent, Yunusobod 6-3-8, tel: 224-45-78");
talaba.Reyting=123.52;
```

Келтирилган мисолда талаба структурасининг сон туридаги майдонларига оддий кўринишда қийматлар берилган, сатр туридаги майдонлар учун стрспй функцияси орқали қиймат бериш амалга оширилган.

Структура туридаги объектнинг хотирадан қанча жой эгаллаганлигини сизеоф функцияси (оператори) орқали аниқлаш мумкин:

```
int i=sizeof(Talaba);
```

Айрим ҳолларда структура майдонлари ўлчамини битларда аниқлаш орқали эгалланадиган хотирани камайтириш мумкин. Бунинг учун структура майдони қуйидагича эълон қилинади:

```
<maydon nomi> : <o'zgarmas ifoda>
```

Бу ерда <майдон номи>- майдон тури ва номи, <ўзгармас ифода>- майдоннинг битлардаги узунлиги. Майдон тури бутун турлар бўлиши керак (int, long, unsigned, char).

Агар фойдаланувчи структуранинг майдони фақат 0 ва 1 қийматини қабул қилишини билса, бу майдон учун бир бит жой ажратиши мумкин (бир байт ёки икки байт ўрнига). Хотирани тежаш эвазига майдон устида амал бажаришда разрядли арифметикани қўллаш зарур бўлади.

Мисол учун сана-вақт билан боғлиқ структурани яратишнинг иккита вариантини кўрайлик. Структура йил, ой, кун, соат, минут ва секунд

майдонларидан иборат бўлсин ва уни қуйидагича аниқлаш мумкин:

```
struct Sana_vaqt
{
    unsigned short Yil;
    unsigned short Oy;
    unsigned short Kun;
    unsigned short Soat;
    unsigned short Minut;
    unsigned short Sekund;
};
```

Бундай аниқлашда Сана\_вақт структураси хотирада 6 майдон\*2 байт=12 байт жой эгаллайди. Агар эътибор берилса структурада ортиқча жой эгалланган ҳолатлар мавжуд. Масалан, йил учун қиймати 0 сонидан 99 сонигача қиймат билан аниқланиши этарли (масалан, 2008 йилни 8 қиймати билан ифодалаш мумкин). Шунинг учун унга 2 байт эмас, балки 7 бит ажратиш этарли. Худди шундай ой учун 1..12 қийматларини ифодалашга 4 бит жой этарли ва ҳакоза.

Юқорида келтирилган чекловлардан кейин сана-вақт структурасини тежамли вариантини аниқлаш мумкин:

```
struct Sana_vaqt2
{
    unsigned Yil:7;
    unsigned Oy:4;
    unsigned Kun:5;
    unsigned Soat:6;
    unsigned Minut:6;
    unsigned Sekund:6;
};
```

Бу структура хотирадан 5 байт жой эгаллайди.

### **23.2. Структура функция аргументи сифатида**

Структуралар функция аргументи сифатида ишлатилиши мумкин. Бунинг учун функция прототипида структура тури кўрсатилиши керак бўлади. Масалан, талаба ҳақидаги берилганларни ўз ичига олувчи Талаба

структураси туридаги берилганларни Talaba\_Manzili() функциясига параметр сифатида бериш учун функция прототипи қуйидаги кўринишда бўлиши керак:

```
void Talaba_Manzili(Talaba);
```

Функцияга структурани аргумент сифатида узатишга мисол сифатидаги дастурнинг матни:

```
#include <iostream>
#include <string.h>
using namespace std;
struct Talaba
{
    char FISH[30];
    unsigned int Tug_yil;
    unsigned int Kurs;
    char Yunalish[50];
    float Reyting;
    unsigned char Jinsi[5];
    char Manzil[50];
    bool status;
};
void Talaba_Manzili(Talaba);
int main()
{
    Talaba talaba;
    talaba.Kurs=2;
    talaba.Tug_yil=1988;
    strcpy(talaba.FISH,"Abdullaev A.A.");
    strcpy(talaba.Yunalish,
    "Informatika va Axborot texnologiyalari");
    strcpy(talaba.Jinsi,"Erk");
    strcpy(talaba.Manzil,
    "Toshkent, Yunusobod 6-3-8, tel: 224-45-78");
    talaba.Reyting=123.52;
    Talaba_Manzili(talaba);
```



```

return 0;
}
void Talaba_Manzili(Talaba t)
{
    cout<<"Talaba FIO: "<<t.FIO<<endl;
    cout<<"Manzili: "<<t.Manzil<<endl;
}

```

Дастур бош функциясида талаба структураси аниқланиб, унинг майдонларига қийматлар берилади. Кейин талаба структураси Talaba\_Manzili() функциясига аргумент сифатида узатилади. Дастур ишлаши натижасида экранга қуйидаги маълумотлар чоп этилади.

```

Talaba FIO: Abdullaev A.A.
Manzili: Toshkent, Yunusobod 6-3-8, tel: 224-45-78

```

### 23.3. Структуралар массиви

Ўз-ўзидан маълумки, структура туридаги ягона берилган билан ечиш мумкин бўлган масалалар доираси жуда тор ва аксарият ҳолатларда, қўйилган масала структуралар мажмуасини ишлатишни талаб қилади. Бу турдаги масалаларга берилганлар базасини қайта ишлаш масалалари деб қараш мумкин.

Структуралар массивини эълон қилиш худди стандарт массивларни эълон қилишдек, фарқи массив тури ўрнида фойдаланувчи томонидан аниқланган структура турининг номи ёзилади. Масалан, талабалар ҳақидаги берилганларни ўз ичига олган массив яратиш эълони қуйидагича бўлади:

```

const int n=25;
Talaba talabalar[n];

```

Структуралар массивининг элементларига мурожаат одатдаги массив элементларига мурожаат усуллари орқали, ҳар бир элементнинг майдонларига мурожаат эса ‘.’ орқали амалга оширилади.

Қуйидаги дастурда гуруҳидаги ҳар бир талаба ҳақидаги берилганларни клавиатурадан киритиш ва гуруҳ талабаларини фамилия, исми ва шарифини чоп қилинади.

```

#include <iostream>
using namespace std;

```

```

#include <conio.h>
const int n=3;
struct Talaba
{
char FISH[30];
unsigned int Tug_yil;
unsigned int Kurs;
char Yunalish[50];
float Reyting;
char Jinsi[6];
char Manzil[50];
bool status;
};
void Talaba_Kiritish(Talaba t[]);
void Talabalar_FISH(Talaba t[]);
int main(int argc, char* argv[])
{
Talaba talabalar[n];
Talaba_Kiritish(talabalar);
Talabalar_FISH(talabalar);
return 0;
}
void Talabalar_FISH(Talaba t[])
{
for(int i=0; i<n; i++)
cout<<t[i].FISH<<endl;
}
void Talaba_Kiritish(Talaba t[])
{
for(int i=0; i<n; i++)
{
cout<<i+1<<"-talaba malumotlarini kiriting:"
<<endl;
cout<<" Talaba FISH :";

```

```

cin.getline(t[i].FISh,30);
cout<<" Kurs:";
cin>>t[i].Kurs;
cout<<" Reyting bali:";
cin>>t[i].Reyting;cout<<" Tug''ilgan yili:";
cin>>t[i].Tug_yil;
cout<<" Ta'lim yo'nalishi:";
cin.getline(t[i].Yunalish,50);
cout<<" Jinsi(erkak,ayol):";
cin.getline(t[i].Jinsi,6);
cout<<" Yashash manzili:";
cin.getline(t[i].Manzil,50);
}
}

```

#### 23.4. Структураларга кўрсаткич

Структура элементларига кўрсаткичлар орқали мурожаат қилиш мумкин. Бунинг учун структурага кўрсаткич ўзгарувчиси эълон қилиниши керак. Масалан, юқорида келтирилган мисолда Талаба структурасига кўрсаткич қуйидагича эълон қилинади:

```
Talaba * k_talaba;
```

Кўрсаткич орқали аниқланган структура элементларига мурожаат «.» билан эмас, балки «->» воситасида амалга оширилади:

```
cout<<k_talaba ->FISh;
```

Структураларни кўрсаткич ва адресни олиш (&) воситасида функция аргументи сифатида узатиш мумкин. Қуйида келтирилган дастур бўлагида структурани Talaba\_Kiritish() функциясига кўрсаткич орқали, Talabalar\_FISh() функциясига эса адресни олиш воситасида узатишга мисол келтирилган.

```

...
void Talaba_Kiritish(Talaba *t);
void Talabalar_FISh(Talaba & t);
int main( )
{

```

```

    Talaba * k_talaba;
    k_talaba=(Talaba*)malloc(n*sizeof(Talaba));
    Talaba_Kiritish(k_talaba);
    Talabalar_FISh(*k_talaba);
    return 0;
}
void Talabalar_FISh(Talaba & t)
{
    for(int i=0; i<n; i++)
    {cout<<(&t+i)->FISh<<endl;}
}
void Talaba_Kiritish(Talaba *t)
{
    for(int i=0; i<n; i++)
    {
        cout<<i+1<<"-talaba malumotlarini kiriting:"
            <<endl;
        cout<<" Talaba FISh :";
        cin.getline((t+i)->FISh,30);
        cout<<" Kurs:";
        cin>>(t+i)->Kurs;
        ...
    }
}

```

Шунга эътибор бериш керакки, динамик равишда ҳосил қилинган структуралар массиви элементи бўлган структуранинг майдониға мурожаатда «\*» белгиси қўлланилмайди.

**Масала.** Футбол жамоалари ҳақидаги маълумотлар - жамоа номи, айна пайтдаги ютуқлар, дуранг ва мағлубиятлар сонлари, ҳамда рақиб дарвозасига киритилган ва ўз дарвозасидан ўтказиб юборилган тўплар сонлари билан берилган. Футбол жамоаларининг турнир жадвали чоп қилинсин. Жамоаларни жадвалда тартиблашда қуйидаги қоидаларға амал қилинсин:

1) жамоалар тўплаган очколарини камайиши бўйича тартибла-ниши керак;

2) агар жамоалар тўплаган очколари тенг бўлса, улардан нисбатан кўп ғалабага эришган жамоа жадвалда юқори ўринни эгаллайди;

3) агар иккита жамоанинг тўплаган очколари ва ғалабалар сони тенг бўлса, улардан нисбатан кўп тўп киритган жамоа жадвалда юқори ўринни эгаллайди.

Жамоа ҳақидаги берилганлар структура кўринишида, жадвал эса структура массиви сифати аниқланади:

```
struct Jamoa
{
    string Nomi;
    int Yutuq, Durang, Maglub, Urgan_tup, Utkazgan_tup;
    int Uyin, Ochko;
};
```

Бу ерда Уйин майдони Ютуқ, Дуранг ва Мағлуб майдонлар йиғиндиси, жамоа тўплаган очколар -  $Очко=3*Ютуқ+1*Дуранг$  кўринишида аниқланади. Жамоалар массиви Очко, Ютуқ ва Урган\_туп майдонлари бўйича тартибланади.

Дастур матни:

```
struct Jamoa
{
    string Nomi;
    int Yutuq, Durang, Maglub, Urgan_tup, Utkazgan_tup;
    int Uyin, Ochko;
};
const nom_uzunligi=10;
int jamoalar_soni;
Jamoa * Jamoalar_Jadvali()
{
    char *jm_nomi=(char*)malloc(nom_uzunligi+1);
    cout<<" Jamoalar soni: ";
    cin>>jamoalar_soni;
    Jamoa * jm=new Jamoa[jamoalar_soni];
    for(int i=0; i<jamoalar_soni; i++)
    {
```

```

cin.ignore();
cout<<i+1<<"-jamo'a ma'lumotlari:\n";
cout<<" Nomi: ";
cin.getline(jm_nomi,nom_uzunligi);
while(strlen(jm_nomi)<nom_uzunligi)
    strcat(jm_nomi," ");
jm[i].Nomi.assign(snomi);
cout<<" Yutuqlar soni: ";
cin>> jm[i].Yutuq;
cout<<" Duranglar soni: ";
cin>>jm[i].Durang;
cout<<" Mag'lubiyatlar soni: ";
cin>>jm[i].Maglub;
cout<<" Raqib darvozasiga urilgan to'plar soni: ";
cin>>jm[i].Urgan_tup;
cout<<" O'z darvozasigan o'tkazgan to'plar soni: ";
cin>>jm[i].Utkazgan_tup;
jm[i].Uyin=jm[i].Yutuq+jm[i].Durang + jm[i].Maglub;
jm[i].Ochko=jm[i].Yutuq*3 +jm[i].Durang;
}
free(snomi);
return jm;
}
void Utkazish(Jamo'a & jamoa1, const Jamoa & jamoa2)
{
    jamoa1.Nomi=jamoa2.Nomi;
    jamoa1.Yutuq=jamoa2.Yutuq;
    jamoa1.Durang=jamoa2.Durang;
    jamoa1.Maglub=jamoa2.Maglub;
    jamoa1.Urgan_tup=jamoa2.Urgan_tup;
    jamoa1.Utkazgan_tup=jamoa2.Utkazgan_tup;
    jamoa1.Uyin=jamoa2.Uyin;
    jamoa1.Ochko=jamoa2.Ochko;
}

```

```

Jamoalar * Jadvalni_Tartiblash(Jamoalar * jm)
{
    bool urin_almashdi=true;
    for(int i=0;i<jamoalar_soni-1 && urin_almashdi; i++)
    {
        Jamoalar Vaqtincha;
        urin_almashdi=false;
        for(int j=i; j<jamoalar_soni-1; j++)
        {
            // j-jamoaning ochkosi (j+1)- jamoa ochkosidan katta
            // bo'lsa, takrorlashning keyingi qadamiga o'tilsin.
            if(jm[j].Ochko>jm[j+1].Ochko) continue;
            //j va (j+1)-jamoalarning ochkolari teng va j-jamoa
            // yutuqlari (j+1)- jamoa yutuqlaridan ko'p bo'lsa,
            // takrorlashning keyingi qadamiga o'tilsin.
            if(jm[j].Ochko==jm[j+1].Ochko &&
                jm[j].Yutuq>jm[j+1].Yutuq) continue;
            //j va (j+1)-jamoalarning ochkolari va yutuqlar soni
            // teng va j-jamoa urgan to'plar soni (j+1)- jamoa
            //urgan to'plardan ko'p bo'lsa, takrorlashning
            //keyingi qadamiga o'tilsin.
            if(jm[j].Ochko==jm[j+1].Ochko &&
                jm[j].Yutuq==jm[j+1].Yutuq &&
                jm[j].Urgan_tup>jm[j+1].Urgan_tup) continue;
            //yuqoridagi shartlarning birortasi ham bajarilmasa,
            //j va (j+1)-jamoalar o'rinlari almashtirilsin.
            urin_almashdi=true;
            Utkazish(Vaqtincha,jm[j]);
            Utkazish(jm[j],jm[j+1]);
            Utkazish(jm[j+1], Vaqtincha);
        }
    }
    return jm;
}

```

```

    void Jadavlni_Chop_Qilish(const Jamoa *jm)
    {
char pr=' ';
cout<<"      FUTBOL JAMOALARINING TURNIR JADVALI\n" ;
cout<<"-----\n";
cout<<"|  JAMOA   | O | Y | D | M |UrT|O'T|OCHKO|\n";
cout<<"-----\n";
for(int i=0; i<jamoalar_soni; i++)
{
    cout<<"|"<<jm[i].Nomi.substr(0,10);cout<<' |';
    if(jm[i].Uyin<10)cout<<pr;cout<<jm[i].Uyin<<" |";
    if(jm[i].Yutuq<10)cout<<pr;cout<<jm[i].Yutuq<<" |";
    if(jm[i].Durang<10)cout<<pr;
    cout<<jm[i].Durang<<" |";
    if(jm[i].Maglub<10)cout<<pr;
    cout<<jm[i].Maglub<<" |";
    if(jm[i].Urgan_tup<10)cout<<pr;
    cout<<jm[i].Urgan_tup<<" |";
    if(jm[i].Utkazgan_tup<10)cout<<pr;
    cout<<jm[i].Utkazgan_tup<<" |";
    if(jm[i].Ochko<10)cout<<pr;
    cout<<jm[i].Ochko<<" |"<<endl;
}
cout<<"-----\n";
}
int main()
{
    Jamoa *jamoas;
    jamoas=Berilganlarni_kiritish();
    jamoas=Jadvalni_Tartiblash(jamoas);
    Jadvalni_Chop_Qilish(jamoas);
    return 0;
}

```



Дастур бош функция ва куйидаги вазифаларни бажарувчи тўртта функциядан ташкил топган:

1) `Jamoa * Jamoalar_Jadvali()` - жамоалар ҳақидаги берилганларни сақлайдиган `Jamoa` структураларидан ташкил топган динамик массив яратади ва унга оқимдан ҳар бир жамоа берилганларни ўқиб жойлаштиради. Ҳосил бўлган массивга кўрсаткични функция натижаси сифа-тида қайтаради;

2) `Jamoa * Jadvalni_Tartiblash(Jamoa * jm)` - аргумент орқали кўрса-тилган массивни масала шарти бўйича тартиблайди ва шу массивга кўрсаткични қайтаради;

3) `void Utkazish(Jamoa & jamoa1, const Jamoa & jamoa2)` - жамоа2 структурасидаги майдонларни жамоа1 структурасига ўтказди. Бу функция `Jadvalni_Tartiblash()` функциясидан массивдаги иккита структурани ўзаро ўринларини алмаштириш учун чақирилади;

4) `void Jadavlni_Chop_Qilish(const Jamoa *jm)` - аргументда берилган массивни турнир жадвали қолипидида чоп қилади.

Учта жамоа ҳақида маълумот берилганда дастур ишлашининг натижаси куйидагича бўлиши мумкин:

#### ФУТБОЛ ЖАМОАЛАРИНИНГ ТУРНИР ЖАДВАЛИ

```
-----
|  ЖАМОА   | О | Й | Д | М | Урт|ЎТ|ОЧКО|
-----
| Бунёдкор | 20 | 15 | 3 | 2 | 30 | 10 | 48 |
| Пахтакор | 20 | 11 | 5 | 4 | 20 | 16 | 38 |
| Нефтчи   | 20 | 8  | 5 | 7 | 22 | 20 | 29 |
-----
```

### 23.5. Бирлашмалар ва улар устида амаллар

Бирлашмалар хотиранинг битта соҳасида (битта адрес бўйича) ҳар хил турдаги бир нечта берилганларни сақлаш имконини беради.

Бирлашма эълони унион калит сўзи, ундан кейин идентификатор ва блок ичида ҳар хил турдаги элементлар эълонидан иборат бўлади, масалан:

```
union Birlashma
{
    int n;
```

```

unsigned long N;
char Satr[10];
};

```

Бирлашманинг бу эълонида компилятор томонидан Бирлашма учун унинг ичидаги энг кўп жой эгалловчи элементнинг - Сатр сатрининг ўлчамида, яъни 10 байт жой ажратилади. Вақтнинг ҳар бир momentiда бирлашмада, эълон қилинган майдонларнинг фақат биттасининг туридаги берилган мавжуд деб ҳисобланади. Юқоридаги мисолда Бирлашма устида амал бажарилишида унинг учун ажратилган хотирада ёки инт туридаги н ёки унсигнед лонг туридаги Н ёки Сатр сатр қиймати жойлашган деб ҳисобланади.

Бирлашма майдонларига худди структура майдонларига мурожаат қилгандек ‘.’ орқали мурожаат қилинади.

Структуралардан фарқли равишда бирлашма эълонида фақат унинг биринчи элементига бошланғич қиймат бериш мумкин:

```

union Birlashma
{
    int n;
    unsigned long N;
    char Satr[10];
}
birlashma={25};

```

Бу мисолда бирлашма бирлашмасининг н майдони бошланғич қиймат олган ҳисобланади.

Бирлашма элементи сифатида структуралар келиши мумкин ва улар одатда берилганни «бўлақларга» ажратиш ёки «бўлақлардан» яхлит берилганни ҳосил қилиш учун хизмат қилади. Мисол учун сўзни байтларга, байтларни тетрадаларга (4 битга) ажратиш ва қайтадан бирлаштириш мумкин.

Қуйида байтни катта ва кичик ярим байтларга ажратишда бирлашма ва структурадан фойдаланилган дастурни матни келтирилган.

```

#include <iostream>
using namespace std;
union BCD
{
    unsigned char bayt;

```

```

struct
    {
        unsigned char lo:4;
        unsigned char hi:4;
    } bin;
} bcd;
int main()
{
    bcd.bayt=127;
    cout<<"\n Katta yarim bayt : "<<(int)bcd.bin.hi;
    cout<<"\n Kichik yarim bayt: "<<(int)bcd.bin.lo;
    return 0;
}

```

Дастур бош функциясида BCD бирлашмасининг байт ўлчамида байт майдонига 127 қиймати берилади ва унинг катта ва кичик ярим байтлари чоп этилади.

Дастур ишлаши натижасида экранга қуйидаги натижалар чиқади:

```

Katta yarim bayt : 7
Kichik yarim bayt: 15

```

**Масала.** Ҳақиқий турдаги соннинг компьютер хотирасидаги ички кўринишини чоп қилиш. Ҳақиқий сон флоат турида деб ҳисобланади ва у хотирада 4 байт жой эгаллайди (1-иловага қаранг). Қўйилган масалани ечиш учун бирлашма хусусиятдан фойдаланилади, яъни хотиранинг битта адресига ҳақиқий сон ва белгилар массиви жойлаштирилади. Ҳақиқий сон хотирага ўқилиб, белгилар массивининг ҳар бир элементининг (байтининг) иккилик кўриниши чоп этилади.

Дастур матни:

```

#include <iostream>
using namespace std;
const unsigned char bitlar_soni=7;
const unsigned char format=sizeof(float);
void Belgi_2kodi(unsigned char blg);
union Son_va_Belgi
{

```

```

float son;
unsigned char belgi[format];
};
int main()
{
    Son_va_Belgi son_va_belgi;
    cin>>son_va_belgi.son;
    for(int b=format-1; b>=0; b--)
        Belgi_2kodi(son_va_belgi.belgi[b]);
    return 0;
}
void Belgi_2kodi(unsigned char blg)
{
    unsigned char l0000000=128;
    for(int i=0;i<=bitlar_soni;i++)
    {
        if(blg&l0000000)cout<<'1';
        else cout<<'0';
        blg=blg<<1;
    }
    cout<<' ';
}

```

Дастурда `Son_va_Belgi` бирлашмасини эълон қилиш орқали флоат туридаги `x` ўзгарувчисини ва `float` тури форматининг байтлардаги узунлигидаги белгилардан иборат белги массивини хотиранинг битта жойига жойлашувига эришилади. Бош функцияда бирлашма туридаги `son_va_belgi` ўзгарувчиси эълон қилинади ва унинг `x` майдонига клавиатурадан ҳақиқий сон ўқилади. Кейин белгилар массивидаги ҳар бир элементнинг иккилик коди чоп этилади. Иккилик кодни чоп этиш 8 марта байтни 7-разрядидаги сонни чоп этиш ва байт разрядларини биттага чапга суриш орқали амалга оширилади. Шунга эътибор бериш керакки, белгилар массивидаги элементларнинг иккилик кодларини чоп қилиш ўнгдан чап томонга бажарилган. Бунга сабаб, сон ички форматидаги байтларнинг хотирада «кичик байт - кичик адресда» қоидасига кўра жойлашувидир.

Дастурга -8.5 сони киритилса, экранда

11000001 00001000 00000000 00000000

кўринишидаги иккилик сонлари кетма-кетлиги пайдо бўлади.

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим

1. Кўрсаткичда массивлардан фойдаланиш.
2. Ичма-ич кўрсаткичлар.
3. Кўрсаткичлар ва функциялар.
4. Кўрсаткич майдонларини киритишда манипуляторлардан фойдаланиш.

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
структура	Struct	бир ёки ҳар хил турдаги берилганларни жамланмаси
бирлашма	Union	майдонларига умумий жой ажратиладиган тузилма
cout		экранга чиқариш оқими

cin		экрандан киритиш оқими
Функция	Function	программинг алоҳида бўлаги, асосий қисм томонидан чақириб ишлатилади
Параметр	Parametr	функция ишлаши учун керак берилганлар
аргумент	Argument	функцияга параметрига жўнатиладиган қиймат
Ўзгарувчи	variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
Идентификатор	identifier	катта ва кичик лотин ҳарфлари, рақамлар ва таг чизиқ ('_') белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги
NULL		мавжуд бўлмаган қиймат
Кўрсаткич	pointer	қийматлари адрес бўлган ўзгарувчилар

### Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур талаба структурасидан фойдаланиб, 1-курсда ўқийдиган талабалар рўйхатини чоп қилиши керак. Аммо талаба маълумотларини киритишда айрим майдонларга қиймат бериш имкони йўқолиб қолмоқда. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб дастур кодини тақдим этинг (индивидуал ҳолда).

### Назорат саволлари

1. Структура деб нимага айтилади?
2. Бирлашма деб нимага айтилади?
3. Бирлашма ва структуранинг фарқи нимада?
4. Структура майдонлари қандай турдарда бўлиши мумкин?
5. Структура майдони ўлчамлари хажмини қандай кўринишда аниқ кўрсатиш мумкин?
6. Структурани функция аргументи сифатида ишлатишга мисол келтиринг.
7. Структуралар массиви қандай эълон қилинади?
8. Структура майдонларига қандай мурожаат қилиш мумкин?
9. Структурага кўрсаткич қандай ишлатилади?
10. Ичма-ич структуралар қандай ишлатилади?

### Тест саволлари

1. Структура майдонларига қандай мурожаат қилинади?
  - a. ‘.’ белгиси орқали
  - b. ‘->.’ орқали
  - c. ‘::’ орқали
  - d. ‘-’ белгиси орқали
2. Кўрсаткич орқали эълон қилинган структура майдонларига қандай мурожаат қилинади?
  - a. ‘.’ белгиси орқали
  - b. ‘->.’ орқали
  - c. ‘::’ орқали
  - d. ‘-’ белгиси орқали

3. Структура майдони ўлчами аниқ кўрсатилган кўриниши қайси жавобда тўғри берилган?

- a. unsigned int Yil:7;
- b. unsigned int Yil-7;
- c. unsigned int Yil.7;
- d. unsigned int Yil::7;

4. Структуралар массиви тўғри кўрсатилган жавобни белгиланг.

- a. const int n=25;  
Talaba talabalar[n];
- b. const int n=25;  
Talaba [n]talabalar;
- c. const int n=25;  
Talaba talabalar\*[n];
- d. const int n=25;  
Talaba talabalar{ {n} };

5. Структурага кўрсаткич тўғри эълон қилинган жавобни топинг.

- a. Talaba \* k\_talaba;
- b. Talaba -> k\_talaba;
- c. Talaba & k\_talaba;
- d. Talaba >> k\_talaba;

6. Қуйидаги дастур қисми бажарилиши натижасида экранга нима чиқади.

```
struct Vaqt{unsigned int h,m,s};  
int main(int argc, char* argv[])  
{  
    Vaqt a={0,59,59},b={0,5,5}; a.m=b.h+b.s;  
    cout<<a.h<<":"<<a.m<<":"<<a.s<<endl;  
    return 0;  
}
```

- a. 0:5:59
- b. 0:5:61
- c. 0:6:4
- d. 0:5:44;



7. Қуйидаги дастур қисми бажарилиши натижасида экранга нима чиқади.

```
struct Racional{int surat,maxraj};
void main(int argc, char* argv[])
{
Racional a={52,33}, b={12,4};
b.surat=(a.surat+4)/a.maxraj;
a.surat=(a.surat+b.maxraj)/b.surat;
cout<<b.surat<<"/"<<a.maxraj<<endl;
}
```

- a. 1/33
- b. 41/33
- c. 5/33
- d. 52/33

8. Қуйидаги дастур қисми бажарилиши натижасида экранга нима чиқади.

```
struct A{int x;};
void f(int a){a=a*a;};
int main(int argc, char* argv[])
{
A a; a.x=5;
f(a.x); cout<<a.x<<endl;
return 0;
}
```

- a. 5
- b. 6
- c. 4
- d. 8

9. Қуйидаги дастур қисми бажарилиши натижасида экранга нима чиқади.

```
struct Vaqt{unsigned int soat,minut,sekund;};
int main(int argc, char* argv[])
{
Vaqt a={2,34,11},b={1,12,5}; b.minut=a.minut+9;
a.sekund=b.sekund+a.minut;
cout<<b.minut<<":"<<a.sekund<<":"<<b.soat<<endl;
return 0;
}
```

- a. 43:39:1
- b. 44:25:61
- c. 30:26:44
- d. 50:55:44;

10. Қуйидаги дастур қисми бажарилиши натижасида экранга нима чиқади.

```
struct Ishchi{unsigned int Yoshi; float Vazni;};  
int main(int argc, char* argv[])  
{  
    Ishchi Eshmat={20,77.8},Toshmat={25,100};  
    if(Eshmat.Vazni<Toshmat.Vazni) cout<<"Eshmat!"<<endl;  
    else cout<<"Toshmat!"<<endl;  
    return 0;  
}
```

- a. Eshmat!
- b. Toshmat!
- c. Eshmat! Toshmat!
- d. Toshmat! Eshmat!

## Маъруза 25-26. Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ўқиш ва ёзиш функциялари.

### Режа:

1. Файл тушунчаси.
2. Матн ва бинар файллар
3. Ўқиш-ёзиш оқимлари. Стандарт оқимлар
4. Белгиларни ўқиш-ёзиш функциялари
5. Сатрларни ўқиш - ёзиш функциялари
6. Форматли ўқиш ва ёзиш функциялари

Таянч атамалар: *include, file, файл, Файл кўрсаткичи, бинар, матн, оқим, cin, if, for, return, ўзгарувчи, идентификатор, Функция, Параметр, аргумент, NULL, EOF.*

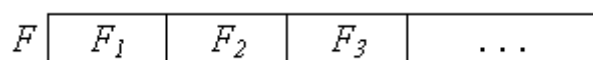
### 25.1. Файл тушунчаси

C++ тилидаги стандарт ва фойдаланувчи томонидан аниқланган турларнинг муҳим хусусияти шундан иборатки, уларнинг олдиндан аниқланган миқдордаги чекли элементлардан иборатлигидир. Ҳатто берилганлар динамик аниқланганда ҳам, оператив хотиранинг (уюмнинг) амалда чекланганлиги сабабли, бу берилганлар миқдори юқоридан чегараланган элементлардан иборат бўлади. Айрим бир тадбиқий масалалар учун олдиндан берилганнинг компоненталари сонини аниқлаш имкони йўқ. Улар масалани ечиш жараёнида аниқланади ва этарлича катта ҳажмда бўлиши мумкин. Иккинчи томондан, дастурда эълон қилинган ўзгарувчиларнинг қийматлари сифатида аниқланган берилганлар фақат дастур ишлаш пайтидагина мавжуд бўлади ва дастур ўз ишини тугатгандан кейин йўқолиб кетади. Агар дастур янгидан ишга туширилса, бу берилганларни янгидан ҳосил қилиш зарур бўлади. Аксарият тадбиқий масалалар эса берилганларни доимий равишда сақлаб туришни талаб қилади. Масалан, корхона ходимларининг ойлик маошини ҳисобловчи дастурда ходимлар рўйхатини, штат ставкалари ва ходимлар томонидан олинган маошлар ҳақидаги маълумотларни

доимий равишда сақлаб туриш зарур. Бу талабларга файл туридаги объектлар (ўзгарувчилар) жавоб беради.

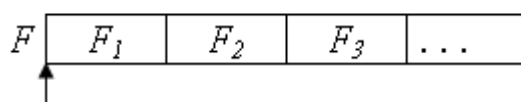
*Файл* - бу бир хил турдаги қийматлар жойлашган ташқи хотирадаги номланган соҳадир.

Файлни, бошида кетма-кет равишда жойлашган ёзувлар (масалан, муסיқа) билан тўлдирилган ва охири бўш бўлган этарлича узун магнит тасмасига ўхшатиш мумкин.



1-расм. Файл тасвири

1-расмда F- файл номи, F<sub>1</sub>, F<sub>2</sub>, F<sub>3</sub> - файл элементлари (компоненталари). Худди янги муסיқани тасма охирига қўшиш мумкин бўлгандек, янги ёзувлар файл охирига қўшилиши мумкин.



2-расм. Файл кўрсаткичи

Яна бир муҳим тушунчалардан бири файл кўрсаткичи тушунчасидир. *Файл кўрсаткичи* - айна пайтда файлдан ўқиладиган ёки унга ёзиладиган жойни (ёзув ўрнини) кўрсатиб туради, яъни файл кўрсаткичи кўрсатиб турган жойдан битта ёзувни ўқиш ёки шу жойга янги ёзувни жойлаштириш мумкин. 2-расмда файл кўрсаткичи файл бошини кўрсатмоқда.

Файл ёзувларига мурожаат кетма-кет равишда амалга оширилади: n-ёзувга мурожаат қилиш учун n-1 ёзувни ўқиш зарур бўлади. Шунини таъкидлаб ўтиш зарурки, файлдан ёзувларни ўқиш жараёни қисман «автоматлашган», унда i - ёзувни ўқилгандан кейин, кўрсаткич навбатдаги i+1 ёзув бошига кўрсатиб туради ва шу тарзда ўқишни давом эттириш мумкин (массивлардагидек индексни ошириш шарт эмас).

Файл - бу берилганларни сақлаш жойидир ва шу сабабли унинг ёзувлари устида тўғридан-тўғри амал бажариб бўлмайди. Файл ёзуви устида амал бажариш учун ёзув қиймати оператив хотирага мос турдаги ўзгарувчига ўқилиши керак. Кейинчалик, зарур амаллар шу ўзгарувчи устида бажарилади ва керак бўлса натижалар яна файлга ёзилиши мумкин.

Операцион система нуқтаи-назаридан файл ҳисобланган ҳар қандай файл C++ тили учун *моддий файл* ҳисобланади. MS DOS учун моддий файллар <файл номи>.<файл кенгайтмаси> кўринишидаги «8.3» форматидаги

сатр (ном) орқали бериледи. Файл номлари сатр ўзгармаслар ёки сатр ўзгарувчиларида берилиши мумкин. MS DOS қоидаларига кўра файл номи тўлиқ бўлиши, яъни файл номининг бошида адрес қисми бўлиши мумкин: ” C:\\USER\\Misol.cpp”, “D:\\matn.txt”.

C++ тилида *мантқиқий файл* тушунчаси бўлиб, у файл туридаги ўзгарувчини англатади. Файл туридаги ўзгарувчиларга бошқа турдаги ўзгарувчилар каби қиймат бериш оператори орқали қиймат бериб бўлмайди. Бошқача айтганда файл туридаги ўзгарувчилар устида ҳеч қандай амал аниқланмаган. Улар устида бажариладиган барча амаллар функциялар воситасида бажарилади.

Файллар билан ишлаш қуйидаги босқичларни ўз ичига олади:

- файл ўзгарувчиси албатта дискдаги файл билан боғланади;
- файл очилади;
- файл устида ёзиш ёки ўқиш амаллари бажарилади;
- файл ёпилади;
- файл номини ўзгартириш ёки файлни дискдан ўчириш амалларини бажарилиши мумкин.

## **25.2. Матн ва бинар файллар**

C++ тили C тилидан ўқиш-ёзиш амалини бажарувчи стандарт функциялар кутубхонасини ворислик бўйича олган. Бу функциялар <стдио.х> сарлавҳа файлида эълон қилинган. Ўқиш-ёзиш амаллари файллар билан бажарилади. Файл матн ёки бинар (иккилик) бўлиши мумкин.

*Матн файл* - ASCII кодидаги белгилар билан берилганлар мажмуаси. Белгилар кетма-кетлиги сатрларга бўлинган бўлади ва сатрнинг тугаш аломати сифатида CR (кареткани қайтариш ёки ‘\r’) LF (сатрни ўтказиш ёки ‘\n’) белгилар жуфтлиги ҳисобланади. Матн файлдан берилганларни ўқишда бу белгилар жуфтлиги битта CR белгиси билан алмаштирилади ва аксинча, ёзишда CR белгиси иккита CR ва LF белгиларига алмаштирилади. Файл охири #26 (^Z) белгиси билан белгиланади.

Матн файлга бошқача таъриф бериш ҳам мумкин. Агар файлни матн таҳририда экранга чиқариш ва ўқиш мумкин бўлса, бу матн файл. Клавиатура ҳам компьютерга фақат матнларни жўнатади. Бошқача айтганда дастур томонидан экранга чиқариладиган барча маълумотларни stdout номидаги

матн файлига чиқарилмоқда деб қараш мумкин. Худди шундай клавиатурадан ўқиладиган ҳар қандай берилганларни матн файлидан ўқилмоқда деб ҳисобланади.

Матн файлларининг компоненталари *сатрлар* деб номланади. Сатрлар узлуксиз жойлашиб, турли узунликда ва бўш бўлиши мумкин. Фараз қилайлик, Т матн файли 4 сатрдан иборат бўлсин:

1- satr#13#10	2- satr uzunroq #13#10	#13#10	4-satr#13#10#26
---------------	------------------------	--------	-----------------

3-расм. Тўртта сатрдан ташкил топган матн файли

Матнни экранга чиқаришда сатр охиридаги #13#10 бошқарув белгилари жуфтлиги курсорни кейинги қаторга туширади ва уни сатр бошига олиб келади. Бу матн файл экранга чоп этилса, унинг кўриниши қуйидагича бўлади:

```
1- satr[13][10]
2- satr uzunroq[13][10]
[13][10]
4- satr[13][10]
[26]
```

Матндаги [n] - n- кодли бошқарув белгисини билдиради. Одатда матн таҳрирлари бу белгиларни кўрсатмайди.

*Бинар файллар* - бу оддийгина байтлар кетма-кетлиги. Одатда бинар файллардан берилганларни фойдаланувчи томонидан бевосита «кўриш» зарур бўлмаган ҳолларда ишлатилади. Бинар файллардан ўқиш-ёзишда байтлар устида ҳеч қандай конвертатсия амаллари бажарилмайди.

### 25.3. Ўқиш-ёзиш оқимлари. Стандарт оқимлар

Оқим тушунчаси берилганларни файлга ўқиш-ёзишда уларни белгилар кетма-кетлиги ёки оқими кўринишида тасаввур қилишдан келиб чиққан. Оқим устида қуйидаги амалларни бажариш мумкин:

- оқимдан берилганлар блокини оператив хотирага ўқиш;
- оператив хотирадаги берилганлар блокини оқимга чиқариш;
- оқимдаги берилганлар блокини янгилаш;
- оқимдан ёзувни ўқиш;
- оқимга ёзувни чиқариш.

Оқим билан ишлайдиган барча функциялар буферли, форматлашган ёки форматлашмаган ўқиш-ёзишни таъминлайди.

Дастур ишга тушганда ўқиш-ёзишнинг қуйидаги стандарт оқимлар очилади:

`stdin` - ўқишнинг стандарт воситаси;

`stdout` - ёзишнинг стандарт воситаси;

`stderr` - хатолик ҳақида хабар беришнинг стандарт воситаси;

`stdprn` - қоғозга чоп қилишнинг стандарт воситаси;

`stdaux` - стандарт ёрдамчи қурилма.

Келишув бўйича `stdin` - фойдаланувчи клавиатураси, `stdout` ва `stderr` - терминал (экран), `stdprn` - принтер билан, ҳамда `stdaux` - компьютер ёрдамчи портларига боғланган ҳисобланади. Берилганларни ўқиш-ёзишда `stderr` ва `stdaux` оқимидан бошқа оқимлар буферланади, яъни белгилар кетма-кетлиги оператив хотиранинг буфер деб номланувчи соҳасида вақтинча жамланади. Масалан, белгиларни ташқи қурилмага чиқаришда белгилар кетма-кетлиги буферда жамланади ва буфер тўлгандан кейингина ташқи қурилмага чиқарилади.

Ҳозирдаги операцион системаларда клавиатура ва дисплейлар матн файллари сифатида қаралади. Ҳақиқатдан ҳам берилганларни клавиатурадан дастурга киритиш (ўқиш) мумкин, экранга эса чиқариш (ёзиш) мумкин. Дастур ишга тушганда стандарт ўқиш ва ёзиш оқимлари ўрнига матн файлларни тайинлаш орқали бу оқимларни қайта аниқлаш мумкин. Бу ҳолатни *ўқишни (ёзишни) қайта адреслаш рўй берди* дейилади. Ўқиш учун қайта адреслашда ‘<’ белгисидан, ёзиш учун эса ‘>’ белгисидан фойдаланилади. Мисол учун `gauss.exe` бажарилувчи дастур берилганларни ўқишни клавиатурадан эмас, балки `massiv.txt` файлидан амалга ошириш зарур бўлса, у буйруқ сатрида қуйидаги кўринишда юкланиши зарур бўлади:

```
gauss.exe < massiv.txt
```

Агар дастур натижасини `natija.txt` файлига чиқариш зарур бўлса

```
gauss.exe > natija.txt
```

сатри ёзилади.

Ва ниҳоят, агар берилганларни `massiv.txt` файлидан ўқиш ва натижани `natija.txt` файлига ёзиш учун

```
gauss.exe < massiv.txt > natija.txt
```

буйруқ сатри терилади.

Умуман олганда, бир дастурнинг чиқиш оқимини иккинчи дастурнинг кириш оқими билан боғлаш мумкин. Буни *конвейрли жўнатиш* дейилади. Агар иккита junat.exe дастурси qabul.exe дастурсига берилганларни жўнатиши керак бўлса, у ҳолда улар ўртасига ‘|’ белги қўйиб ёзилади:

```
junat.exe | qabul.exe
```

Бу кўринишдаги дастурлар ўртасидаги конвейрли жўнатишни операцион системанинг ўзи таъминлайди.

#### 25.4. Белгиларни ўқиш-ёзиш функциялари

Белгиларни ўқиш-ёзиш функциялари макрос кўринишида амалга оширилган.

getc () макроси тайинланган оқимдан навбатдаги белгини қайтаради ва кириш оқими кўрсаткичини кейинги белгини ўқишга мослаган ҳолда оширади. Агар ўқиш муваффақиятли бўлса getc () функцияси ишорасиз инт кўринишидаги қийматни, акс ҳолда EOF қайтаради. Ушбу функция прототипи қуйидагича:

```
int getc(FILE * stream)
```

EOF идентификатор макроси

```
#define EOF(-1)
```

кўринишида аниқланган ва ўқиш-ёзиш амалларида файл охирини белгилаш учун хизмат қилади. EOF қиймати ишорали чар турида деб ҳисобланади. Шу сабабли ўқиш-ёзиш жараёнида унсигнед чар туридаги белгилар ишлатилса, EOF макросини ишлатиб бўлмайди.

Навбатдаги мисол getc() макросини ишлатишни намоён қилади.

```
#include <iostream>
```

```
using namespace std;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    char ch;
```

```
    cout<<"Belgini kiriting: ";
```

```
    ch=getc(stdin);
```

```
    cout<<"Siz " <<ch<<" belgisini kiritdingiz.\n";
```



```
    return 0;
}
```

getc() макроси аксарият ҳолатларда stdin оқими билан ишлатилганлиги сабабли, унинг getc(stdin) кўринишига эквивалент бўлган int getchar() макроси аниқланган. Юқоридаги мисолда «ch=getc(stdin);» каторини «ch=getchar();» катори билан алмаштириш мумкин.

Белгини оқимга чиқариш учун putchar() макроси аниқланган ва унинг прототипи

```
int puts(const char *s);
```

кўринишида аниқланган. putchar() функцияси stream номи билан берилган оқимга с белгини чиқаради. Функция қайтарувчи қиймати сифатида инт турига айлантирилган с белги бўлади. Агар белгини чиқаришда хатолик рўй берса EOF қайтарилади.

putchar() функциясини стандарт stdout оқими билан боғланган ҳолати - putchar(c, stdout) учун putchar(c) макроси аниқланган.

## 25.5. Сатрларни ўқиш - ёзиш функциялари

Оқимдан сатрни ўқишга мўлжалланган gets() функциясининг прототипи

```
char * gets(char *s);
```

кўринишида аниқланган. gets() функцияси стандарт оқимдан сатрни ўқийди ва уни с ўзгарувчисига жойлаштиради. Жойлаштириш пайтида оқимдаги '\n' белгиси '\0' белгиси билан алмаштирилади. Бу функцияни ишлатишда ўқиладиган сатрнинг узунлиги s сатр учун ажратилган жой узунлигидан ошиб кетмаслигини назорат қилиш керак бўлади.

puts() функцияси

```
int puts(const char *s);
```

кўринишида бўлиб, у стандарт оқимга аргументда кўрсатилган сатрни чиқаради. Бунда сатр охирига янги сатрга ўтиш белгиси '\n' қўшилади. Агар сатрни оқимга чиқариш муваффақиятли бўлса puts() функцияси манфий бўлмаган сонни, акс ҳолда EOF қайтарди.

Сатрни ўқиш-ёзиш функцияларини ишлатишга мисол тариқасида кўйидаги дастурни келтириш мумкин.

```
#include <stdio.h>
int main()
```

```

{
    char *s;
    puts("Satrni kiriting: ");
    gets(s);
    puts("Kiritilgan satr: ");
    puts(s);
    return 0;
}

```

## 25.6. Форматли ўқиш ва ёзиш функциялари

Форматли ўқиш ва ёзиш функциялари - `scanf()` ва `printf()` функциялари C тилидан ворислик билан олинган. Бу функцияларни ишлатиш учун «`stdio.h`» сарлавҳа файлини дастурга қўшиш керак бўлади.

Форматли ўқиш функцияси `scanf()` қуйидаги прототипга эга:

```
int scanf(const char * <format>[<adres>, ...])
```

Бу функция стандарт оқимдан берилганларни форматли ўқишни амалга оширади. Функция, кириш оқимидаги майдонлар кетма-кетлиги кўринишидаги белгиларни бирма-бир ўқийди ва ҳар бир майдонни `<format>` сатрида келтирилган формат аниқлаштирувчисига мос равишда форматлайди. Оқимдаги ҳар бир майдонга формат аниқлаштирувчиси ва натижа жойлашадиган ўзгарувчининг адреси бўлиши шарт. Бошқача айтганда, оқимдаги майдон (ажратилган белгилар кетма-кетлиги) кўрсатилган форматдаги қийматга акслантирилади ва ўзгарувчи билан номланган хотира бўлагига жойлаштирилади (сақланади). Функция оқимдан берилганларни ўқиш жараё-нини «тўлдирувчи белгини» учратганда ёки оқим тугаши натижасида тўхтатиши мумкин. Оқимдан берилганларни ўқиш мувафакқиятли бўлса, функция мувафакқиятли айлантирилган ва хотирага сақланган майдонлар сонини қайтаради. Агар ҳеч бир майдонни сақлаш имкони бўлмаган бўлса, функция 0 қийматини қайтаради. Оқим охирига келиб қолганда (файл ёки сатр охирига) ўқишга ҳаракат бўлса, функция EOF қийматини қайтаради.

Форматлаш сатри - `<format>` белгилар сатри бўлиб, у учта тоифага бўлинади:

- тўлдирувчи белгилар;
- тўлдирувчи белгилардан фарқли белгилар;
- формат аниқлаштирувчилари.

*Тўлдирувчи-белгилар* – бу пробел, ‘\t’, ‘\n’ белгилари. Бу белгилар форматлаш сатридан ўқилади, лекин сақланмайди.

*Тўлдирувчи белгилардан фарқли белгилар* – бу қолган барча ASCII белгилари, ‘%’ белгисилан ташқари. Бу белгилар форматлаш сатридан ўқилади, лекин сақланмайди.

1–жадвал. Формат аниқлаштирувчилари ва уларнинг вазифаси

<b>Компонента</b>	<b>Бўлиши шарт ёки йўқ</b>	<b>Вазифаси</b>
[*]	Йўқ	Навбатдаги кўриб чиқиладиган майдон қийматини ўзгарувчига ўзлаштирмаслик белгиси. Кириш оқимидаги майдон кўриб чиқилади, лекин ўзгарувчида сақланмайди.
[<kenglik>]	Йўқ	Майдон кенглигини аниқлаштирувчиси. Ўқиладиган белгиларнинг максимал сонини аниқлайди. Агар оқимда тўлдирувчи белги ёки алмаштирилмайдиган белги учраси функция нисбатан кам сондаги белгиларни ўқиши мумкин.
[F N]	Йўқ	Ўзгарувчи кўрсаткичининг (адресининг) модификатори: F – far pointer; N- near pointer
[h l L]	Йўқ	Аргумент турининг модификатори. <tur belgisi> билан аниқланган ўзгарувчининг қисқа (short - h) ёки узун (long – l, L) кўринишини аниқлайди.
<tur belgisi>	Ҳа	Оқимдаги белгиларни алмаштириладиган тур белгиси

*Формат аниқлаштирувчилари* – оқим майдонидаги белгиларни кўриб чиқиш, ўқиш ва адреси билан берилган ўзгарувчилар турига мос равишда алмаштириш жараёнини бошқаради. Ҳар бир формат аниқлаштирувчисига битта ўзгарувчи адреси мос келиши керак. Агар формат аниқлаштирувчилари сони ўзгарувчилардан кўп бўлса, натижа нима бўлишини олдиндан айтиб бўлмайди. Акс ҳолда, яъни ўзгарувчилар сони кўп бўлса, ортиқча ўзгарувчилар инобатга олинмайди.

Формат аниқлаштирувчиси қуйидаги кўринишга эга:

`% [*][<kenglik>] [F|N] [h|l|L] <tur belgisi>`

Формат аниқлаштирувчиси `%` белгисидан бошланади ва ундан кейин 1–жадвалда келтирилган шарт ёки шарт бўлмаган компоненталар келади.

2–жадвал. Алмаштириладиган тур аломати белгилари

Тур аломати	Кутилаётган қиймат	Аргумент тури
<b>Сон туридаги аргумент</b>		
d, D	Ўнлик бутун	int * arg yoki long * arg
E, e	Сузувчи нуқтали сон	float * arg
F	Сузувчи нуқтали сон	float * arg
G, g	Сузувчи нуқтали сон	float * arg
O	Саккизлик сон	int * arg
O	Саккизлик сон	long * arg
I	Ўнлик, саккизлик ва ўн олтилик бутун сон	int * arg
I	Ўнлик, саккизлик ва ўн олтилик бутун сон	long * arg
U	Ишорасиз ўнлик сон	Unsigned int * arg
U	Ишорасиз ўнлик сон	Unsigned long * arg
x	Ўн олтилик сон	int * arg
X	Ўн олтилик сон	int * arg
<b>Белгилар</b>		
S	Сатр	char * arg (белгилар массиви)
C	Белги	char * arg (белги учун майдон кенглиги берилиши мумкин (масалан, %4s). Н белгидан ташкил топган белгилар массивига кўрсаткич: char arg [N])
%	'%' белгиси	Ҳеч қандай алмаштиришлар бажарилмайди, '%' белгиси сақланади.

Кўрсаткичлар		
N	int * arg	%n аргументигача муваффақиятли ўқилган белгилар сони, айнан шу инт кўрсаткичи бўйича адресда сақланади.
P	YYYY:ZZZZ ёки ZZZZ кўринишидаги ўн олтилик	Объектга кўрсаткич (far* ёки near*).

Оқимдаги белгилар бўлагини алмаштириладиган тур аломатининг қабул қилиши мумкин бўлган белгилар 2-жадвалда келтирилган.

3-жадвал. Формат аниқлаштирувчилари ва уларнинг вазифаси

Компонента	Бўлиши шарт ёки йўқ	Вазифаси
[bayroq]	Йўқ	Байроқ белгилари. Чиқарилаётган қийматни чапга ёки ўнга текислашни, соннинг ишорасини, ўнлик каср нуқтасини, охирдаги нолларни, саккизлик ва ўн олтилик сонларнинг аломатларни чоп этишни бошқаради. Масалан, ‘-‘ байроғи қийматни ажратилган ўринга нисбатан чапдан бошлаб чиқаришни ва керак бўлса ўнгдан пробел билан тўлдиришни билдиради, акс ҳолда чап томондан пробеллар билан тўлдиради ва давомига қиймат чиқарилади.
[<kenglik>]	Йўқ	Майдон кенглигини аниқлаштирувчиси. Чиқариладиган белгиларнинг минимал сонини аниқлайди. Зарур бўлса қиймат ёзилишидан ортган жойлар пробел билан тўлдирилади.
[.<xona>]	Йўқ	Аниқлик. Чиқариладиган белгиларнинг максимал сонини кўрсатади. Сондаги рақамларнинг минимал сонини.
[F N h l L]	Йўқ	Ўлчам модификатори. Аргументнинг қисқа (short - h) ёки узун (long - l,L) кўринишини, адрес турини аниқлайди.
<tur belgisi>	Ҳа	Аргумент қиймати алмаштириладиган тур аломати белгиси

Форматли ёзиш функцияси printf() қуйидаги прототипга эга:

```
int printf(const char * <format>[, <argument>, ...])
```

Бу функция стандарт оқимга форматлашган чиқаришни амалга оширади.

Функция аргументлар кетма-кетлигидаги ҳар бир аргумент қийматини қабул қилади ва унга <формат> сатридаги мос формат аниқлаштирувчисини қўллайди ва оқимга чиқаради.

4–жадвал. printf() функциясининг алмаштириладиган тур белгилари

Тур аломати	Кутилаётган қиймат	Чиқиш формати
Сон қийматлари		
D	Бутун сон	Ишорали ўнлик бутун сон
I	Бутун сон	Ишорали ўнлик бутун сон
O	Бутун сон	Ишорасиз саккизлик бутун сон
U	Бутун сон	Ишорасиз ўнлик бутун сон
x	Бутун сон	Ишорасиз ўн олтилик бутун сон (a,b,c,d,e,f белгилари ишлатилади)
X	Бутун сон	Ишорасиз ўн олтилик бутун сон (A,B,C,D,E,F белгилари ишлатилади)
F	Сузувчи нуқтали сон	[-]dddd.dddd кўринишидаги сузувчи нуқтали сон
E	Сузувчи нуқтали сон	[-]d.dddd ёки e[+/-]ddd кўринишидаги сузувчи нуқтали сон
G	Сузувчи нуқтали сон	Кўрсатилган аниқликка мос e ёки f шаклидаги сузувчи нуқтали сон
E, G	Сузувчи нуқтали сон	Кўрсатилган аниқликка мос e ёки f шаклидаги сузувчи нуқтали сон. E формат учун 'E' чоп этилади.
Белгилар		
S	Сатрга кўрсаткич	0-белгиси учрамагунча ёки кўрсатилган аниқликка эришилмагунча белгилар оқимга чиқарилади.
C	Белги	Битга белги чиқарилади
%	Ҳеч нима	'%' белгиси оқимга чиқарилади.

Кўрсаткичлар		
N	int ko‘rsatkich (int* arg)	%n аргументигача муваффақиятли чиқарилган белгилар сони, айнан шу инт кўрсаткичи бўйича адресда сақланади.
P	Кўрсаткич	Аргументни YYY:ZZZ ёки ZZZZ кўринишидаги ўн олтилик сонга айлантириб оқимга чиқаради.

Ҳар бир формат аниқлаштирувчисига битта ўзгарувчи адреси мос келиши керак. Агар формат аниқлаштирувчилари сони ўзгарувчилардан кўп бўлса, натижада нима бўлишини олдиндан айтиб бўлмайди. Акс ҳолда, яъни ўзгарувчилар сони кўп бўлса, ортиқча ўзгарувчилар инобатга олинмайди. Агар оқимга чиқариш муваффақиятли бўлса, функция чиқарилган байтлар сонини қайтаради, акс ҳолда EOF.

printf() функциясининг <format> сатри аргументларни алмаштириш, форматлаш ва берилганларни оқимга чиқариш жараёнини бошқаради ва у икки турдаги объектлардан ташкил топади:

- оқимга ўзгаришсиз чиқариладиган оддий белгилар;
- аргументлар рўйхатидаги танланадиган аргументга қўлланиладиган формат аниқлаштирувчилари.

Формат аниқлаштирувчиси қуйидаги кўринишга эга:

```
% [<bayroq>][<.kenglik>] [.<xona>][F|N|h|l|L] <tur belgisi>
```

Формат аниқлаштирувчиси ‘%’ белгисидан бошланади ва ундан кейин 3–жадвалда келтирилган шарт ёки шарт бўлмаган компоненталар келади.

Алмаштириладиган тур белгисининг қабул қилиши мумкин бўлган белгилар 4- жадвалда келтирилган.

Берилганлар қийматларини оқимдан ўқиш ва оқимга чиқаришда scanf() ва printf() функцияларидан фойдаланишга мисол:

```
#include <stdio.h>
int main()
{
    int bson, natija;
    float hson;
```

```

char blg, satr[81];
printf("\nButun va suzuvchi nuqtali sonlarni,");
printf("\nbelgi hamda satrni kiriting\n");
natija=scanf("%d %f %c %s", &bson, &hson,&blg,satr);
printf("\nOqimdan %d ta qiymat o'qildi ",natija);
printf("va ular quyidagilar:");
printf("\n %d %f %c %s \n",bson, hson, blg, satr);
return 0;
}

```

Дастур фойдаланувчидан бутун ва сузувчи нуқтали сонларни, белги ва сатрни киритишни сўрайди. Бунга жавобан фойдаланувчи томонидан

10 12.35 A Satr

қийматлари киритилса, экранга

Oqimdan 4 ta qiymat o'qildi va ular quyidagilar:

10 12.35 A Satr

сатрлари чоп этилади.

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим

1. Файллар ҳақида маълумотлар.
2. Стандарт оқимлар кўринишлари.
3. Белгиларни билан ишлаш функциялари.
4. Сатрлар билан ишлаш функциялари.



## Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
Файл	File	бу бир хил турдаги қийматлар жойлашган ташқи хотирадаги номланган соҳадир
Файл кўрсаткичи	File pointer	айни пайтда файлдан ўқиладиган ёки унга ёзиладиган жойни (ёзув ўрнини) кўрсатиб туради
cout		экранга чиқариш оқими
cin		экрандан киритиш оқими
Функция	Function	программинг алоҳида бўлаги, асосий қисм томонидан чақириб ишлатилади
Параметр	Parametr	функция ишлаши учун керак берилганлар
аргумент	Argument	функцияга параметрига жўнатилган қиймат
Ўзгарувчи	variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
Идентификатор	identifier	катта ва кичик латин ҳарфлари, рақамлар ва тағчизик (‘_’) белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги
NULL		мавжуд бўлмаган қиймат

EOF		#define EOF(-1) кўринишида аниқланган макрос
Кўрсаткич	pointer	қийматлари адрес бўлган ўзгарувчилар

### Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур талаба структурасидан фойдаланиб, 1-курсда ўқийдиган талабалар рўйхатини чоп қилиши керак. Аммо талаба маълумотларини киритишда айрим майдонларга қиймат бериш имкони йўқолиб қолмоқда. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб дастур кодини тақдим этинг (индивидуал ҳолда).

### Назорат саволлари

1. Файл нима?
2. Файл кўрсаткичи деб нимага айтилади?
3. Матн файл ва бинар файлнинг фарқи нимада?
4. Форматли ўқиш учун қандай функция ишлатилади ва унинг синтаксиси қандай?
5. Форматли ёзиш учун қандай функция ишлатилади ва унинг синтаксиси қандай?
6. Қандай ўқиш оқимларини биласиз?
7. Қандай ёзиш оқимларини биласиз?
8. Белгиларни ўқиш учун қайси функциялар ишлатилади?
9. Белгиларни ёзиш учун қайси функциялар ишлатилади?
10. Сатрларни ўқиш учун қайси функциялар ишлатилади
11. Сатрларни ёзиш учун қайси функциялар ишлатилади

## Тест саволлари

1. Файл деб нимага айтилади?
  - a. бир хил турдаги қийматлар жойлашган ташқи хотирадаги номланган соҳадир
  - b. ҳар хил турдаги қийматлар жойлашган ташқи хотирадаги номланган соҳадир
  - c. бир хил турдаги қийматлар жойлашган ташқи хотирадаги номланмаган соҳадир
  - d. ҳар хил турдаги қийматлар жойлашган ташқи хотирадаги номланмаган соҳадир
  
2. *Файл кўрсаткичи ... кўрсатиб туради?*
  - a. айна пайтда файлдан ўқиладиган ёки унга ёзиладиган жойни
  - b. файлдан белги ўқиш жараёнини
  - c. файлдан сатр ўқиш жараёнини
  - d. айна пайтда файлдан ўқиладиган белгининг ASCII жадвалидаги кодини
  
3. Файлдан  $i$  - ёзувни ўқилгандан кейин, кўрсаткич нечанчи ёзувни кўрсатиб туради?
  - a.  $i+1$
  - b.  $i+2$
  - c.  $i-1$
  - d.  $i$
  
4. C++ тилида мантиқий файл бу ...
  - a. файл туридаги ўзгарувчи
  - b. файл жойлашган жой
  - c. файлдаги белгилар сони
  - d. файлнинг кенгайтмаси
  
5. Матн файл - ...
  - a. ASCII кодидаги белгилар билан берилганлар мажмуаси
  - b. C++ да очиш мумкин бўлган ихтиёрий файл
  - c. Рақамлардан ташкил топган файл
  - d. Кенгайтмаси "txt" бўлган файл

6. Форматли ўқиш учун қандай функция ишлатилади?

- a. scanf()
- b. printf()
- c. gets()
- d. puts()

7. Форматли ёзиш учун қандай функция ишлатилади?

- a. scanf()
- b. printf()
- c. gets()
- d. puts()

8. Белгини ўқиш функцияси тўғри кўрсатилган қаторни топинг

- a. gets()
- b. puts()
- c. getc()
- d. putc()

9. Белгини ёзиш функцияси тўғри кўрсатилган қаторни топинг

- a. gets()
- b. puts()
- c. getc()
- d. putc()

10. Тўлдирувчи-белгилар тўғри кўрсатилган жавобни топинг

- a. пробел, '\t', '\n' белгилари
- b. 'a', '\t', '\n' белгилари
- c. Enter, Esc, '\t', '\n' белгилари
- d. 'a', 'b', 'c', 'd' белгилари

## Маъруза 27-28. Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари.

### Режа:

1. Файлдан ўқиш-ёзиш функциялари.
2. Файл кўрсаткичини бошқариш функциялари.
3. Файллар оқимлари билан ишлаш.
4. Файллар оқимларида ўқиш-ёзиш функциялари.

Таянч атамалар: *include*, *Файл*, *Файл кўрсаткичи*, *cout*, *cin*,  
*Функция*, *Параметр*, *Ўзгарувчи*, *Идентификатор*, *FILE*,  
*Кўрсаткич*, *EOF*, *fstream*, *ifstream*, *ofstream*.

### 27.1. Файлдан ўқиш-ёзиш функциялари

Файл оқими билан ўқиш-ёзиш амалини бажариш учун файл оқимини очиш зарур. Бу ишни, прототипи

```
FILE * fopen(const char* filename, const char *mode);
```

кўринишида аниқланган `fopen()` функцияси орқали амалга оширилади. Функция `filename` номи билан файлни очади, у билан оқимни боғлайди ва оқимни идентификатсия қилувчи кўрсаткични жавоб тариқасида қайтаради. Файлни очиш муваффақиятсиз бўлганлигини `fopen()` функциясининг `NULL` қийматли жавоби билдиради.

Параметрлар рўйхатидаги иккинчи - `mode` параметри файлни очиш режимини аниқлайди. У қабул қилиши мумкин бўлган қийматлар куйидаги жадвалда келтирилган.

#### Файл очиш режимлари

Қиймати	Файл очилиш ҳолати тавсифи
r	Файл фақат ўқиш учун очилади
w	Файл ёзиш учун очилади. Агар бундай файл мавжуд бўлса, у қайтадан ёзилади (янгиланади).
a	Файлга ёзувни қўшиш режими. Агар файл мавжуд бўлса, файл унинг охирига ёзувни ёзиш учун очилади, акс ҳолда янги файл яратилади ва ёзиш режимида очилади.

r+	Мавжуд файл ўзгартириш (ўқиш ва ёзиш) учун очилади.
w+	Янги файл яратилиб, ўзгартириш (ўқиш ва ёзиш) учун очилади. Агар файл мавжуд бўлса, ундаги олдинги ёзувлар ўчирилади ва у қайта ёзишга тайёрланади.
a+	Файлга ёзувни қўшиш режими. Агар файл мавжуд бўлса, унинг охирига (EOF аломатидан кейин) ёзувни ёзиш (ўқиш) учун очилади, акс ҳолда янги файл яратилади ва ёзиш режимида очилади.

Матн файли очилаётганлигини билдириш учун файл очилиш режими сатрига ‘t’ белгисини қўшиб ёзиш зарур бўлади. Масалан, матн файл ўзгартириш (ўқиш ва ёзиш) учун очилаётганлигини билдириш учун “rt+” сатри ёзиш керак бўлади. Худди шундай бинар файллар устида ишлаш учун ‘b’ белгисини ишлатиш керак. Мисол учун файл очилишининг “wb+” режими бинар файл янгиланишини билдиради.

Файл ўзгартириш (ўқиш-ёзиш) учун очилганда, берилганларни оқимдан ўқиш, ҳамда оқимга ёзиш мумкин. Бироқ ёзиш амалидан кейин дарҳол ўқиб бўлмади, бунинг учун ўқиш амалидан олдин fseek() ёки rewind() функциялари чақирилиши шарт.

Фараз қилайлик «C:\\USER\\TALABA\\iat1kurs.txt» номли матн файлни ўқиш учун очиш зарур бўлсин. Бу талаб

```
FILE *f=fopen("C:\\USER\\TALABA\\iat1kurs.txt", "r+");
```

ифодасини ёзиш орқали амалга оширалади. Натижада дискда мавжуд бўлган файл дастурда f ўзгарувчиси номи билан айнан бир нарса деб тушунилади. Бошқача айтганда, дастурда кейинчалик f устида бажарилган барча амаллар, дискдаги «iat1kurs.txt» файли устида рўй беради.

Файл оқими билан ишлаш тугагандан кейин у албатта ёпилиши керак. Бунинг учун fclose() функциясидан фойдаланилади. Функция прототипи қуйидаги кўринишга эга:

```
int fclose(FILE * stream);
```

fclose() функцияси оқим билан боғлиқ буферларни тозалайди (масалан, файлга ёзиш кўрсатмалари берилиши натижасида буферда йиғилган берилганларни дискдаги файлга кўчиради) ва файлни ёпади. Агар файлни ёпиш хатоликка олиб келса, функция EOF қийматини, нормал ҳолатда 0

қийматини қайтаради.

`fgetc()` функцияси прототиби

```
int fgetc(FILE *stream);
```

кўринишида аниқланган бўлиб, файл оқимидан белгини ўқишни амалга оширади. Агар ўқиш муваффақиятли бўлса, функция ўқилган белгини инт туридаги ишорасиз бутун сонга айлантиради. Агар файл охирини ўқишга ҳаракат қилинса ёки хатолик рўй берса, функция EOF қийматини қайтаради.

Кўриниб турибдики, `getc()` ва `fgetc()` функциялари деярли бир хил ишни бажаради, фарқи шундаки, `getc()` функцияси белгини стандарт оқимдан ўқийди. Бошқача айтганда, `getc()` функцияси, файл оқими стандарт қурилма бўлган `fgetc()` функцияси билан аниқланган макросдир.

`fputc()` функцияси

```
int fputc(int c, FILE *stream);
```

прототиби билан аниқланган. `fputc()` функцияси файл оқимиға аргументда кўрсатилган белгини ёзади (чиқаради) ва у амал қилишида `puts()` функцияси билан бир хил.

Файл оқимидан сатр ўқиш учун

```
char * fgets(char * s, int n, FILE *stream)
```

прототиби билан `fgets()` аниқланган. `fgets()` функцияси файл оқимидан белгилар кетма-кетлигини `s` сатрига ўқийди. Функция ўқиш жараёнини оқимдан `n-1` белги ўқилгандан кейин ёки кейинги сатрга ўтиш белгиси (`'\n'`) учраганда тўхтатади. Охирги ҳолда `'\n'` белгиси ҳам `s` сатрга кўшилади. Белгиларни ўқиш тугагандан кейин `s` сатр охирига, сатр тугаш аломати `'\0'` белгиси кўшилади. Агар сатрни ўқиш муваффақиятли бўлса, функция `s` аргумент кўрсатадиган сатрни қайтаради, акс ҳолда `NULL`.

Файл оқимиға сатрни `fputs()` функцияси ёрдамида чиқариш мумкин. Бу функция прототиби

```
int fputs (const char *s, FILE *stream);
```

кўринишида аниқланган. Сатр охиридаги янги сатрга ўтиш белгиси ва терминаторлар оқимга чиқарилмайди. Оқимга чиқариш муваффақиятли бўлса, функция номанфий сон қайтаради, акс ҳолда EOF.

`feof()` функцияси аслида макрос бўлиб, файл устида ўқиш-ёзиш амаллари бажарилаётганда файл охири белгиси учраган ёки йўқлигини билдиради. Функция

```
int feof(FILE *stream);
```

прототипига эга бўлиб у файл охири белгиси учраса, нолдан фарқли сонни қайтаради, бошқа ҳолатларда 0 қийматини қайтаради.

Қуйида келтирилган мисолда файлга ёзиш ва ўқишга амаллари кўрсатилган.

```
#include <iostream>
using namespace std;
#include <stdio.h>
int main()
{
    char s;
    FILE *in,*out;
    if((in=fopen("D:\\USER\\TALABA.TXT","rt"))==NULL)
    {
        cout<<"Talaba.txt faylini ochilmadi!!\n";
        return 1;
    }
    if((out=fopen("D:\\USER\\TALABA.DBL","wt+"))==NULL)
    {
        cout<<"Talaba.dbl faylini ochilmadi!!\n";
        return 1;
    }
    while (!feof(in))
    {
        char c=fgetc(in);
        cout<<c;
        fputc(c,out);
    }
    fclose(in);
    fclose(out);
    return 0;
}
```



Дастурда «talaba.txt» файли матн файли сифатида ўқиш учун очилган ва у in ўзгарувчиси билан боғланган. Худди шундай, «talaba.dbf» матн файли ёзиш учун очилган ва out билан боғланган. Агар файлларни очиш муваффақиятсиз бўлса, мос хабар берилади ва дастур ўз ишини тугатади. Кейинчалик, токи ин файли охирига этмагунча, ундан белгилар ўқилади ва экранга, ҳамда out файлига чиқарилади. Дастур охирида иккита файл ҳам ёпилади.

**Масала.** Ғалвирли тартиблаш усули.

Берилган  $x$  векторини пуфакча усулида камаймайдиган қилиб тартиблаш куйидагича амалга оширилади: массивнинг  $k$  ўрни элементлари  $x_k$  ва  $x_{k+1}$  ( $k=1, \dots, n-1$ ) солиштирилади. Агар  $x_k > x_{k+1}$  бўлса, у ҳолда бу элементлар ўзаро ўрин алмашади. Шу йўл билан биринчи ўтишда энг катта элемент векторнинг охирига жойлашади. Кейинги қадамда вектор бошидан  $n-1$  ўриндаги элементгача юқорида қайд қилинган йўл билан қолган элементларнинг энг каттаси  $n-1$  ўринга жойлаштирилади ва ҳ.к.

Ғалвирли тартиблаш усули пуфакчали тартиблаш усулига ўхшаш, лекин  $x_k$  ва  $x_{k+1}$  ( $k=1, 2, 3, \dots, n-1$ ) элементлар ўрин алмашгандан кейин «ғалвирдан» ўтказиш амали қўлланилади: чап томондаги кичик элемент имкон қадар чап томонга тартиблаш сақланган ҳолда кўчирилади. Бу усул оддий пуфакчали тартиблаш усулига нисбатан тез ишлайди.

Дастур матни:

```
#include <stdio.h>
#include <alloc.h>
int * Pufakchali_Tartiblash(int*,int);
int main()
{
    char fnomi[80];
    printf("Fayl nomini kiriting:");
    scanf("%s", &fnomi);
    int Ulcham,i=0,* Massiv;
    FILE * f1, *f2;
    if((f1=fopen(fnomi,"rt"))==NULL)
    {
        printf("Xato:%s fayli ochilmadi!",fnomi);
    }
}
```

```

    return 1;
}
fscanf(f1,"%d",&Ulcham);
Massiv=(int *)malloc(Ulcham*sizeof(int));
while(!feof(f1))
fscanf(f1,"%d",&Massiv[i++]);
fclose(f1);
Massiv=Pufakchali_Tartiblash(Massiv,Ulcham);
f2=fopen("natija.txt","wt");
fprintf(f2,"%d%c",Ulcham,' ');
for(i=0; i<Ulcham; i++)
fprintf(f2,"%d%c",Massiv[i],' ');
fclose(f2);
return 0;
}
int * Pufakchali_Tartiblash(int M[],int n)
{
    int almashdi=1, vaqtincha;
    for(int i=0; i<n-1 && almashdi;i++)
    {
        almashdi=0;
        for(int j=0; j<n-i-1;j++)
        if (M[j]>M[j+1])
        {
            almashdi=1;
            vaqtincha=M[j];
            M[j]=M[j+1];
            M[j+1]=vaqtincha;
            int k=j;
            if(k)
            while(k && M[k]>M[k-1])
            {
                vaqtincha=M[k-1];
                M[k-1]=M[k];

```

```

        M[k]=vaqtincha;
        k--;
    }
}
}
return M;
}

```

Дастурда берилганларни оқимдан ўқиш ёки оқимга чиқаришда файлда форматли ўқиш - fscanf() ва ёзиш - fprintf() функцияларидан фойдаланилган. Бу функцияларнинг мос равишда scanf() ва printf() функцияларидан фарқи - улар берилганларни биринчи аргумент сифатида бериладиган матн файлда ўқийди ва ёзади.

Номи фойдаланувчи томонидан киритиладиган f1 файлда бутун сонлар массивининг узунлиги ва қийматлари ўқилади ва тартибланган массив f2 файлга ёзилади.

Векторни тартибланиш Pufakchali\_Tartiblash() функцияси томонидан амалга оширилади. Унга вектор ва унинг узунлиги кирувчи параметр бўлади ва тартибланган вектор функция натижаси сифатида қайтарилади.

Навбатдаги иккита функция файл оқимидан форматлашмаган ўқиш-ёзишни амалга оширишга мўлжалланган.

fread() функцияси қуйидаги прототипга эга:

```

size_t fread(void * ptr, size_t size, size_t n,
             FILE *stream);

```

Бу функция оқимдан ptr кўрсатиб турган буферга, ҳар бири сизе байт бўлган n та берилганлар блоқини ўқийди. Ўқиш муваффақиятли бўлса, функция ўқилган блоқлар сонини қайтаради. Агар ўқиш жараёнида файл охири учраб қолса ёки хатолик рўй берса, функция тўлиқ ўқилган блоқлар сонини ёки 0 қайтаради.

fwrite() функцияси прототипи

```

size_t fwrite(const void*ptr,size_t size,
             size_t n,FILE *stream);

```

кўриниши аниқланган. Бу функция ptr кўрсатиб турган буфердан, ҳар бири сизе байт бўлган n та берилганлар блоқини оқимга чиқаради. Ёзиш муваффақиятли бўлса, функция ёзилган блоқлар сонини қайтаради. Агар ёзиш

жараёнида хатолик рўй берса, функция тўлиқ ёзилган блоклар сонини ёки 0 қайтаради.

## 27.2. Файл кўрсаткичини бошқариш функциялари

Файл очилганда, у билан «stdio.h» сарлавҳа файлида аниқланган FILE структураси боғланади. Бу структура ҳар бир очилган файл учун жорий ёзув ўрнини кўрсатувчи ҳисоблагични - файл кўрсаткичини мос қўяди. Одатда файл очилганда кўрсаткич қиймати 0 бўлади. Файл устида бажарилган ҳар бир амалдан кейин кўрсаткич қиймати ўқилган ёки ёзилган байтлар сонига ошади. Файл кўрсаткичини бошқариш функциялари - fseek(), ftell() ва rewind() функциялари файл кўрсаткичини ўзгартириш, қийматини олиш имконини беради.

ftell() функциясининг прототипи

```
long int ftell(FILE *stream);
```

кўринишида аниқланган бўлиб, аргументда кўрсатилган файл билан боғланган файл кўрсаткичи қийматини қайтаради. Агар хатолик рўй берса функция -1L қийматини қайтаради.

```
int fseek(FILE *stream, long offset, int from);
```

прототипига эга бўлган fseek() функцияси страем файли кўрсаткичини фром жойига нисбатан offset байт масофага суришни амалга оширади. Матн режимидаги оқимлар учун offset қиймати 0 ёки ftell() функцияси қайтарган қиймат бўлиши керак. from параметри қуйидаги қийматларни қабул қилиши мумкин:

SEEK\_SET (=0) - файл боши;

SEEK\_CUR (=1) - файл кўрсаткичининг айна пайтдаги қиймати;

SEEK\_END (=2) - файл охири.

Функция файл кўрсаткичи қийматини ўзгартириш муваффақиятли бўлса, 0 қийматини, акс ҳолда нолдан фарқли қиймат қайтаради.

rewind() функцияси

```
void rewind(FILE *stream);
```

прототипи билан аниқланган бўлиб, файл кўрсаткичини файл бошланишига олиб келади.

Қуйида келтирилган дастурда бинар файл билан ишлаш кўрсатилган.

```
#include <iostream>
```

```

using namespace std;
#include <stdio.h>
#include <string.h>
struct Shaxs
{
    char Familiya[20];
    char Ism[15];
    char Sharifi[20];};
int main()
{
    int n,k;
    cout<<"Talabalar sonini kiriting: "; cin>>n;
    FILE *oqim1,*oqim2;
    Shaxs *shaxs1, *shaxs2, shaxsk;
    shaxs1=new Shaxs[n];
    shaxs2=new Shaxs[n];
    if ((oqim1=fopen("Talaba.dat", "wb+"))==NULL)
    {
        cout<<"Talaba.dat ochilmadi!!!";
        return 1;
    }
    for(int i=0; i<n; i++)
    {
        cout<<i+1<<"- shaxs ma'lumotlarini kiriting:\n";
        cout<<"Familiysi: "; gets(shaxs1[i].Familiya);
        cout<<"Ismi: "; gets(shaxs1[i].Ism);
        cout<<"Sharifi: "; gets(shaxs1[i].Sharifi);
    }
    if (n==fwrite(shaxs1,sizeof(Shaxs),n,oqim1))
        cout<<"Berilganlarni yozish amalga oshirildi!\n";
    else
    {
        cout<<"Berilganlarni yozish amalga oshirilmadi!\n";
        return 3;
    }
}

```

```

}
cout<<" Fayl uzunligi: "<<ftell(oqim1)<<'\\n';
fclose(oqim1);
if((oqim2=fopen("Talaba.dat", "rb+"))==NULL)
{
    cout<<"Talaba.dat o'qishga ochilmadi!!!";
    return 2;
}
if (n==fread(shaxs2,sizeof(Shaxs),n,oqim2))
for(int i=0; i<n; i++)
{
    cout<<i+1<<"- shaxs ma'lumotlari:\\n";
    cout<<"Familiysi: "<<shaxs2[i].Familiya<<'\\n';
    cout<<"Ismi: "<<shaxs2[i].Ism<<'\\n';
    cout<<"Sharifi: "<<shaxs2[i].Sharifi<<'\\n';
    cout<<"*****\\n"; }
else
{
    cout<<"Fayldan o'qish amalga oshirilmadi!\\n" ;
    return 4;
}
do
{
    cout<<"Yo'zuv nomerini kiriting (1.."<<n<<"):";
    cin>>k;
    } while (k<0 && k>n);
k--;
cout<<"Oldingi Familiya: ";
cout<<shaxs2[k].Familiya <<'\\n';
cout<<"Yangi Familiya: ";
gets(shaxs2[k].Familiya);
if (fseek(oqim2, k*sizeof(Shaxs),SEEK_SET))
{
    cout<<"Faylda"<<k+1;

```

```

    cout<<"-yo'zuvga o'tishda xatolik ro'y berdi???\n";
    return 5;
}
fwrite(shaxs2+k,sizeof(Shaxs),1,oqim2);
fseek(oqim2, k*sizeof(Shaxs),SEEK_SET);
fread(&shaxsk,sizeof(Shaxs),1,oqim2);
cout<<k+1<<"- shaxs ma'lumotlari:\n";
cout<<"Familiysi: "<<shaxsk.Familiya<<'\n';
cout<<"Ismi: "<<shaxsk.Ism<<'\n';
cout<<"Sharifi: "<<shaxsk.Sharifi<<'\n';
fclose(oqim2);
delete shaxs1;
delete shaxs2;
return 0;
}

```

Юқорида келтирилган дастурда, олдин «Talaba.dat» файли бинар файл сифатида ёзиш учун очилади ва у oqim1 ўзгарувчиси билан боғланади. Шахс ҳақидаги маълумотни сақловчи н ўлчамли динамик shaxs1 структуралар массиви oqim1 файлига ёзилади, файл узунлиги чоп қилиниб файл ёпилади. Кейин, худди шу файл oqim2 номи билан ўқиш учун очилади ва ундаги берилганлар shaxs2 структуралар массивига ўқилади ва экранга чоп қилинади. Дастурда файлдаги ёзувни ўзгартириш (қайта ёзиш) амалга оширилган. Ўзгартириш қилиниши керак бўлган ёзув тартиб номери фойдаланувчи томонидан киритилади (k ўзгарувчиси) ва shaxs2 структуралар массивидаги мос ўриндаги структуранинг Familiya майдони клавиатурадан киритилган янги сатр билан ўзгартирилади. oqim2 файл кўрсаткичи файл бошидан k\*sizeof(Shaxs) байтга сурилади ва shaxs2 массивнинг k - структураси (shaxs2+k) шу ўриндан бошлаб файлга ёзилади. Кейин oqim2 файли кўрсаткичи ўзгартириш киритилган ёзув бошига қайтарилади ва бу ёзув shaxsk структурасига ўқилади ҳамда экранга чоп этилади.

**Масала.** Ҳақиқий сонлар ёзилган f файли берилган. f файлдаги элементларнинг ўрта арифметигидан кичик бўлган элементлар миқдорини аниқлансин.

Масалани ечиш учун *f* файлини яратиш ва қайтадан уни ўқиш учун очиш зарур бўлади. Яратилган файлнинг барча элементларининг йиғиндиси *s* ўзгарувчисига ҳосил қилинади ва у файл элементлари сонига бўлинади. Кейин *f* файл кўрсаткичи файл бошига олиб келинади ва элементлар қайта ўқилади ва *s* қийматидан кичик элементлар сони - *k* санаб борилади.

Файлни яратиш ва ундаги ўрта арифметикдан кичик сонлар миқдорини аниқлашни алоҳида функция кўринишида аниқлаш мумкин.

Дастур матни:

```
#include <iostream.h>
#include <stdio.h>
#include <string.h>
int Fayl_Yaratish()
{
    FILE * f;
    double x;
    // f fayli yangidan hosil qilish uchun ochiladi
    if ((f=fopen("Sonlar.db1", "wb+"))==NULL)return 0;
    char *satr=new char[10];
    int n=1;
    do
    {cout<<"Sonni kiriting(bo'sh satr tugatish): ";
      gets(satr);
      if(strlen(satr))
      {x=atof(satr);
        fwrite(&x,sizeof(double),n,f);
      }
    } while(strlen(satr));// satr bo'sh bo'lmasa,
                          // takrorlash

    fclose(f);
    return 1;
}
int OAdan_Kichiklar_Soni()
{
    FILE * f;
```



```

double x;
f=fopen("Sonlar.dbl", "rb+");
double s=0; // s - f fayl elementlari yig'indisi
while (!feof(f))
{
    if (fread(&x,sizeof(double),1,f)) s+=x;
}
long sonlar_miqdori=ftell(f)/sizeof(double);
s/=sonlar_miqdori; // s- o'rta arifmetik
cout<<"Fayldagi sonlar o'rta arifmetiki="<<s<<endl;
fseek(f,SEEK_SET,0); // fayl boshiga kelinsin
int k=0;
while (fread(&x,sizeof(x),1,f))
{
    k+=(x<s); //o'rta arifmetikdan kichik elementlar
            //soni
}
fclose(f);
return k;
}
int main()
{
    if(Fayl_Yaratish())
    {
        cout<<"Sonlar.dbl faylidagi\n";
        int OA_kichik=OAdan_Kichiklar_Soni();
        cout<<"O'rta arifmetikdan kichik sonlar miqdori=";
        cout<<OA_kichik;
    }
    else // f faylini yaratish muvaffaqiyatsiz bo'ldi.
        cout<<"Faylini ochish imkoni bo'lmadi!!!";
    return 0;
}

```

Дастурда бош функциядан ташқари иккита функция аниқланган:

`int Fayl_Yaratish()` - дискда «Sonlar.dbl» номли файлни яратади. Агар файлни яратиш муваффақиятли бўлса, функция 1 қийматини, акс ҳолда 0 қийматини қайтаради. Файлни яратишда клавиатурадан сонларнинг сатр кўриниши ўқилади ва сонга айлантирилиб, файлга ёзилади. Агар бўш сатр киритилса, сонларни киритиш жараёни тўхта-тилади ва файл ёпилади;

`int OAdan_Kichiklar_Soni()` - дискдаги «Sonlar.dbl» номли файли ўқиш учун очилади ва файл элементларининг `s` ўрта арифметигидан кичик элементлари сони `k` топилади ва функция натижаси сифатида қайтарилади.

Бош функцияда файлни яратиш муваффақиятли кечганлиги текширилади ва шунга мос хабар берилади.

### **27.3. Файллар оқимлари билан ишлаш.**

C++да файллар оқимлари билан ишлаш учун `fstream` кутубхонаси мавжуд.

`fstream` кутубхонаси файлларни ўқиб олиш учун жавоб берадиган `ifstream` синфига, ҳамда файлга маълумот ёзишга имкон берадиган `ofstream` турларига (синфига) эга.

```
#include <fstream>
```

Бирон-бир файлни ёзиш ёки ўқиш учун очиш учун, `ofstream` турдаги ёки мос ҳолда `ifstream` турдаги ўзгарувчини яратиш керак.

```
ifstream inData;
```

```
ofstream outData;
```

Бундай ўзгарувчини инициаллашда файл номи ўзгарувчи номидан кейин кавс ичида берилган белгилар массиви кўринишида узатилади. Масалан, C дискида жойлашган “text.txt” файлини очиш керак. Бунинг учун коднинг қуйидаги фрагменти қўлланади:

```
ifstream inData (“C:\\text.txt”);
```

```
ofstream outData (“C:\\text.txt”);
```

ёки ўзгарувчи орқали қуйидагича инициаллаш мумкин.

```
char s[20] = “C:\\text.txt”;
```

```
ifstream inData(s);
```

Шунингдек, файлни очиш учун `open()` функциясини ҳам ишлатиш мумкин.

```
ofstream outData;
outData.open("cppfayl.txt");
```

Агар файл ҳам дастурнинг бажарилаётган файли жойлаштирилган папкада бўлса, у холда файлнинг номи тўлиқ кўрсатилмаслиги мумкин (фақат файл номи, унга бориш йўлисиз). Бундан ташқари файл номини тўғридан-тўғри кўрсатиш ўрнига, унинг номидан иборат белгилар массивларини кўрсатиш мумкин.

Агар файллар умумий файл оқими турида яратилса, файлни очиш режими кўрсатилиши керак. Қўшиш тузимида файлни очиш учун очилишда қуйида кўрсатилган иккинчи параметрни кўрсатиш лозим:

```
fstream output_file("cppfayl.txt", ios::app);
```

Бу холда ios::app параметри файлни очиш режимини аниқлайди.

#### Очиш режимлари.

Очиш режими	Вазифаси
ios::app	Файл кўрсаткични файлни охирида жойлаштириб кўшиш режимида файлни очади.
ios::ate	Файл кўрсаткичини файлни охирига жойлаштиради. Ўқиш мумкин эмас, чиқарувчи маълумотлар файлни охирига ёзилади.
ios::in	Қиритиш учун файлни очилишини кўрсатади.
ios::nocreate	Агарда кўрсатилган файл мавжуд бўлмаса, файл яратилмайди ва хато қайтарилади.
ios::noreplace	Агарда файл мавжуд бўлса, очиш операцияси тўхташ ва хатоликни қайтариш лозим.
ios::out	Чиқариш учун файлни очилишини кўрсатади
ios::trunc	Мавжуд бўлган файлни ичидагини олиб ташлайди (кўчиради).
ios::binary	Файлни бинар файл кўринишида очади

Бир нечта режимни биргаликда ишлатиш имкони ҳам мавжуд. Бунда разрядли ёки амалидан фойдаланилади. Қуйида файлни очиш ҳамда уни тозалаб ташлашни олдини олиш учун ios::noreplace режимларидан фойдаланилган:

```
ifstream output_file("cppfayl.txt", ios::out | ios::noreplace);
```

Дастурни тугаллаш учун операция тизими ўзи очган файлларни беркитади. Бироқ, одатга кўра, агар дастурга файл керак бўлмай қолса, уни беркитиши керак. Файлни беркитиш учун дастур, қуйида кўрсатилганидек, close функциясидан фойдаланиши керак:

```
output_file.close();
```

#### 27.4. Файллар оқимларида ўқиш-ёзиш функциялари

Ахборотни файлга ёзиш учун put() функциясидан фойдаланиш мумкин. Бу функция орқали стандарт турдаги яқка ўзгарувчи ёки бирон-бир белгилар массиви узатилади. Белгилар массиви узатилган холда массивдаги ёзилиши керак бўлган белгилар сонини узатиш керак.

Бундан ташқари “<<” операторидан фойдаланиш мумкин. Бу оператордан коднинг битта сатрида турли турдаги қийматларни узатган холда кўп марталаб фойдаланиш мумкин. Сатр хақида гап кетганда, чиқариш сатр охири белгиси, яъни ‘\n’ пайдо бўлишидан олдин амалга оширилади. Белгисиз турга эга бўлган барча ўзгарувчилар олдин белгиларга ўзгартириб олинади.

```
ofstream outData("C:\\text.txt");
char a='M';
outData.put(s);
char s[21]="Bir necha so'zli matn";
outData.put(s,21);
outData << "Oqim belgisi orqali yozish";
int i=100;
outData << "Bir nechta son " << i << " " << 200;
```

Ахборотни файлдан ўқиб олиш учун “>>” оператори ва get функциясидан фойдаланиш мумкин. put функцияси каби, get функцияси ҳам хар қандай ўзгарувчиларнинг стандарт турлари, белгилар массивлари билан ишлай олади. Шунингдек get га хар жихатдан эквивалент бўлган getline функциясини ҳам ишлатиш мумкин.

```
ifstream inData("C:\\text.txt");
char s;
char ss[9];
s=inData.get();
cout << s;
```

```

inData.get(s);
cout << s;
inData.getline(ss,9);
cout << ss;
inData >> ss;
cout << ss;

```

Файл охирини аниқлаш учун, оқим объектининг eof() функциясидан фойдаланиш мумкин. Агар файл охири хали учрамаган бўлса, бу функция 0 қийматини қайтаради, агар файл охири учраса - 1 қийматини қайтаради. while циклидан фойдаланиб, файл охири топилмагунча файл билан бирор иш бажариш қуйидаги кодда келтирилган:

```

while ( ! inData.eof() )
{
// бажариладиган ишлар
}

```

Ушбу холда дастур, eof функцияси ёлғон (0) ни қайтаргунча, циклни бажаришда давом этади.

Массивлар ва тузилмаларни ўқиш ва ёзиш керак бўлса, read ва write функцияларидан фойдаланишлари мумкин. read ва write функцияларидан фойдаланишда маълумотлар ўқиладиган ёки ёзиб олинadиган маълумотлар буферини, шунингдек буфернинг байтларда ўлчанадиган узунлигини кўрсатиш лозим. Бу қуйида кўрсатилганидек амалга оширилади:

```

input_file.read(buffer, sizeof(buffer));
output_file.write(buffer, sizeof(buffer));

```

Масалан, тузилма ичидаги маълумотларни EMPLOYEE.DAT файлига чиқариш учун, write функциясидан фойдаланади:

```

struct employee
{
char name[64];
int age;
float salary;
} worker = { "Djon Doy", 33, 25000.0 };
ofstream emp_file("EMPLOYEE.DAT") ;
emp_file.write((char *) &worker, sizeof(employee));

```

Одатда write функцияси белгилар сатрига кўрсаткич олади. Худди

шундай тарзда read функцияси орқали хизматчи хақидаги ахборотни файлдан ўқиб олиш учун фойдаланади:

```
ifstream emp_file("EMPLOYEE.DAT");
emp_file.read((char *) &worker, sizeof(employee));
cout << worker.name << endl;
cout << worker.age << endl;
cout << worker.salary << endl;
```

### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

### Мустақил таълим

1. Бинар файлларга маълумотларни ёзиш функциялари.
2. Бинар файллардан маълумотларни ўқиш функциялари.
3. Файл оқимларида кўрсаткични бошқариш функциялари.
4. Файл оқимларида хатоларни аниқлаш функциялари.

### Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
Файл	File	бу бир хил турдаги қийматлар жойлашган ташқи хотирадаги номланган соҳадир

Файл кўрсаткичи	File pointer	айни пайтда файлдан ўқиладиган ёки унга ёзиладиган жойни (ёзув ўрнини) кўрсатиб туради
cout		экранга чиқариш оқими
cin		экрандан киритиш оқими
Функция	Function	программинг алоҳида бўлаги, асосий қисм томонидан чақириб ишлатилади
Параметр	Parametr	функция ишлаши учун керак берилганлар
Ўзгарувчи	variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги
Идентификатор	identifier	катта ва кичик латин ҳарфлари, рақамлар ва таг чизиқ ('_') белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлиги
FILE		Файллар билан ишлаш тури
Кўрсаткич	pointer	қийматлари адрес бўлган ўзгарувчилар
EOF		Файлни охирини аниқловчи ўзгарувчи
fstream		Файл оқими
ifstream		Ўқиш файли оқими
ofstream		Ёзиш файли оқими

## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур талаба структураси маълумотларини файлга ёзиши керак. Дастур ишлади, аммо файлга ҳеч нарса ёзилмади. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб турли усулларда дастур кодини тақдим этинг (индивидуал ҳолда).

## Назорат саволлари

1. Файл деб нимага айтилади?
2. C++ тилида файл билан ишловчи қандай турлар мавжуд?
3. FILE\* тури орқали файлни қандай очиш мумкин?
4. Файлни очиш режимлари нима учун керак?
5. Қандай файлни очиш режимлари мавжуд?
6. Файл оқими турида файлни очиш режимларининг қандай кўринишлари мавжуд.
7. Файлни қандай ёпиш мумкин?
8. Файлга маълумотлар қандай ёзилади?
9. Файл бошига қандай қайтиш мумкин?
10. Файл кўрсаткичини қандай бошқариш мумкин?

## Тест саволлари

1. Файл оқими тури тўғри берилган жавобни белгиланг?
  - a. fstream
  - b. int
  - c. FILE\*
  - d. fopen



2. Қуйида келтирилган турларнинг қайси бири файллар билан ишлаш учун ўлжалланган?

- a. FILE\*
- b. double
- c. fopen
- d. double

3. Матн файлни ўқиш учун очиш режими қандай белгиланади?

- a. rt+
- b. rb+
- c. wb+
- d. wt+

4. Бинар файлни ўқиш учун очиш режими қандай белгиланади?

- a. rt+
- b. rb+
- c. wb+
- d. at+

5. Матн файлни ёзиш учун очиш режими қандай белгиланади?

- a. rt+
- b. rb+
- c. wb+
- d. wt+

6. Бинар файлни ёзиш учун очиш режими қандай белгиланади?

- a. rt+
- b. rb+
- c. wb+
- d. at+

7. Файлга сатр ёзиш учун қуйидаги функцияларнинг қайси бири ишлатилади?

- a. fgets()
- b. fputs()
- c. fgetc()
- d. fopen()

8. Файлга белги ёзиш учун қуйидаги функцияларнинг қайси бири ишлатилади?

- a. fgets()
- b. fputc()
- c. fgetc()
- d. feof()

9. Файлдан белги ўқиш учун қуйидаги функцияларнинг қайси бири ишлатилади?

- a. fgets()
- b. fputc()
- c. fgetc()
- d. feof()

10. Файлдан сатр ўқиш учун қуйидаги функцияларнинг қайси бири ишлатилади?

- a. fgets()
- b. fputc()
- c. fgetc()
- d. feof()

## Маъруза 29-30. Динамик тузилмалар

### Режа:

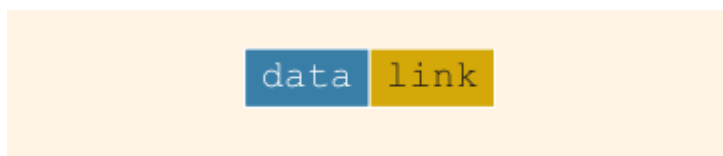
1. Берилганларнинг динамик тузилмалари.
2. Чизиқли рўйхат куриш.
3. Чизиқли рўйхат устида амаллар.
4. Навбат.
5. Стек.

Таянч атамалар: *динамик тузилмалар, чизиқли рўйхат, навбат, стек, head, NULL.*

### 29.1. Берилганларнинг динамик тузилмалари

Берилганлар устида ишлашда уларнинг миқдори қанча бўлиши ва уларга хотирадан қанча жой ажратиш кераклиги олдиндан номаълум бўлиши мумкин. Программа ишлаш пайтида берилганлар учун зарурат бўйича хотирадан жой ажратиш ва уларни кўрсаткичлар билан боғлаш орқали ягона структура ҳосил қилиш жараёни хотиранинг динамик тақсимоти дейилади. Бу усулда ҳосил бўлган берилганлар мажмуасига берилганларнинг динамик структураси дейилади, чунки уларнинг ўлчами программа бажарилишида ўзгариб туради. Дастурлашда динамик структуралардан чизиқли рўйхатлар, стеклар, навбатлар ва бинар дарахтлар ҳосил қилишда нисбатан кўп ишлатилади. Улар бир-биридан элементлар ўртасидаги боғланишлари ва улар устида бажариладиган амаллари билан фарқланади. Ҳар қандай берилганларнинг динамик структураси майдонлардан ташкил топади ва уларнинг айримлари қўшни элементлар билан боғланиш учун хизмат қилади.

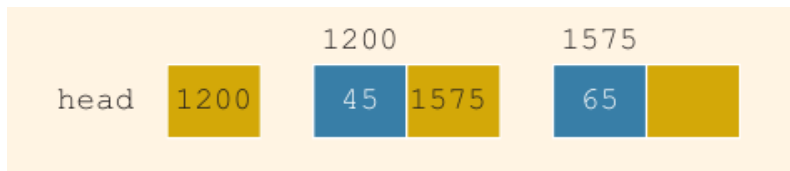
Чизиқли рўйхат тугунлардан (node) ташкил топади. Занжирнинг ҳар бир тугуни иккита майдондан иборат: қиймат майдони (data) ва кейинги элемент билан боғланишни таъминловчи кўрсаткич (link).



Биринчи тугун жойлашган хотиранинг манзили боши (head) ёки биринчиси дейилади.



Чизиқли рўйхатнинг ҳар бир тугуни ўзидан кейинги тугуннинг хотирадаги манзилни кўрсатиб туради. Сўнгги тугун NULL ни кўрсатади. Чунки ундан кейин ҳеч қандай тугун йўқ.



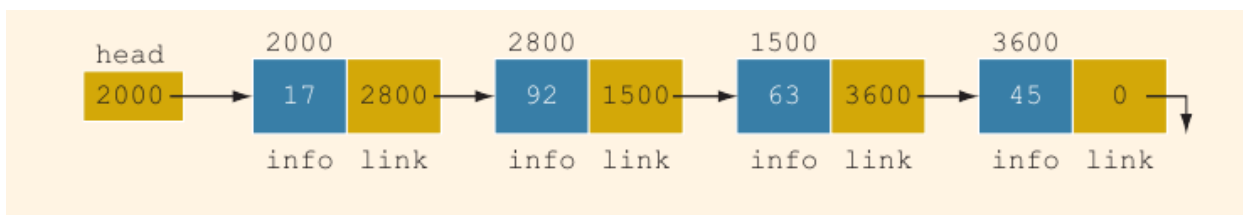
“head” кўрсатиб турган манзил – 1200. Ушбу манзилда 45 қийматига эга тугун жойлашган. Бу тугун эса ўз навбатида хотиранинг 1575 манзилни кўрсатиб турибди.

Бундай чизиқли рўйхатни ҳосил қилишда тузилмалардан фойдаланилади:

```
struct nodeType
{
int info;
nodeType *link;
};
```

Ўзгарувчи эълон қилиш қуйидагича амалга оширилади:

```
nodeType *head;
```

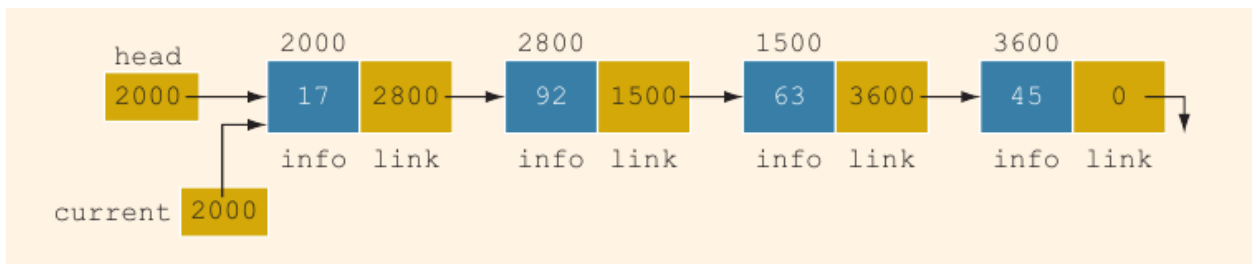


Ушбу кўринишда чизиқли рўйхат қийматлари қуйидагича кўринишда бўлади:

```
head                2000
head->info           17
head->link           2800
head->link->info     92
```

Агар head ни эълон қилгандек current кўрсаткичини эълон қилиб, қуйидагича қиймат юкласак, current ва head битта манзилни кўрсатиб туради:

```
current = head;
```



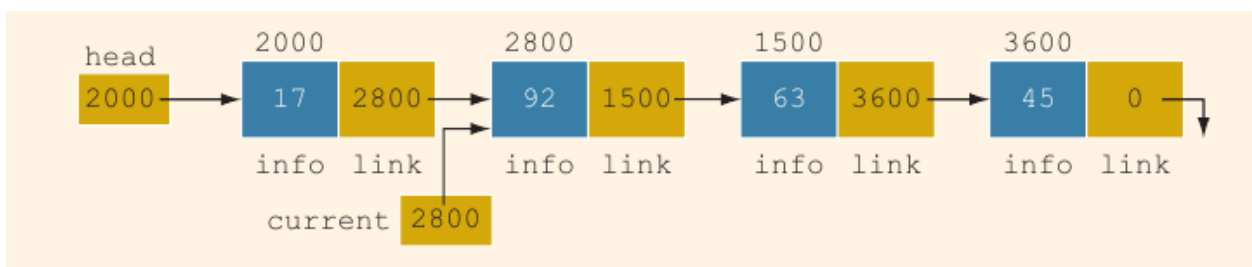
Энди, current орқали тугунлар қийматларини олиш мумкин:

```
current          2000
current->info    17
current->link    2800
current->link->info 92
```

Бундай юклашнинг асосий мақсади, дастур ишлаши мобайнида чизиқли рўйхат бошини йўқотиб қўймаслик. Дастур ишлаши давомида headнинг қиймати ўзгармаслиги керак, акс ҳолда занжирнинг бошини топиб бўлмайди. Чизиқли рўйхат тугунларида ҳаракатланишни current кўрсаткичи таъминлайди. Тугунларда ҳаракатланиш учун қуйидагича қиймат юклаш амалга оширилади:

```
current = current->link;
```

Қиймат юкланиши жараёнида current кўрсаткичининг қиймати current->link қийматини олади, натижада current 2800-хотира манзилини кўрсатиб туради. Ушбу манзилда эса чизиқли рўйхатнинг иккинчи тугуни жойлашган.



```
current          2800
current->info    92
current->link    1500
current->link->info 63
```

Чизиқли рўйхатнинг сўнги элементи гача бориш учун қуйидагича цикл ишлатиш мумкин:

```
current = head;
while (current != NULL)
{
```

```
//бирор бажариладиган амаллар
current = current->link;
}
```

Масалан, чизиқли рўйхатнинг барча элементларини чоп қилиш керак бўлса, қуйидагича дастур матни ишлатилиши мумкин:

```
current = head;
while (current != NULL)
{
cout << current->info << " ";
current = current->link;
}
```

## 29.2. Чизиқли рўйхат қуриш

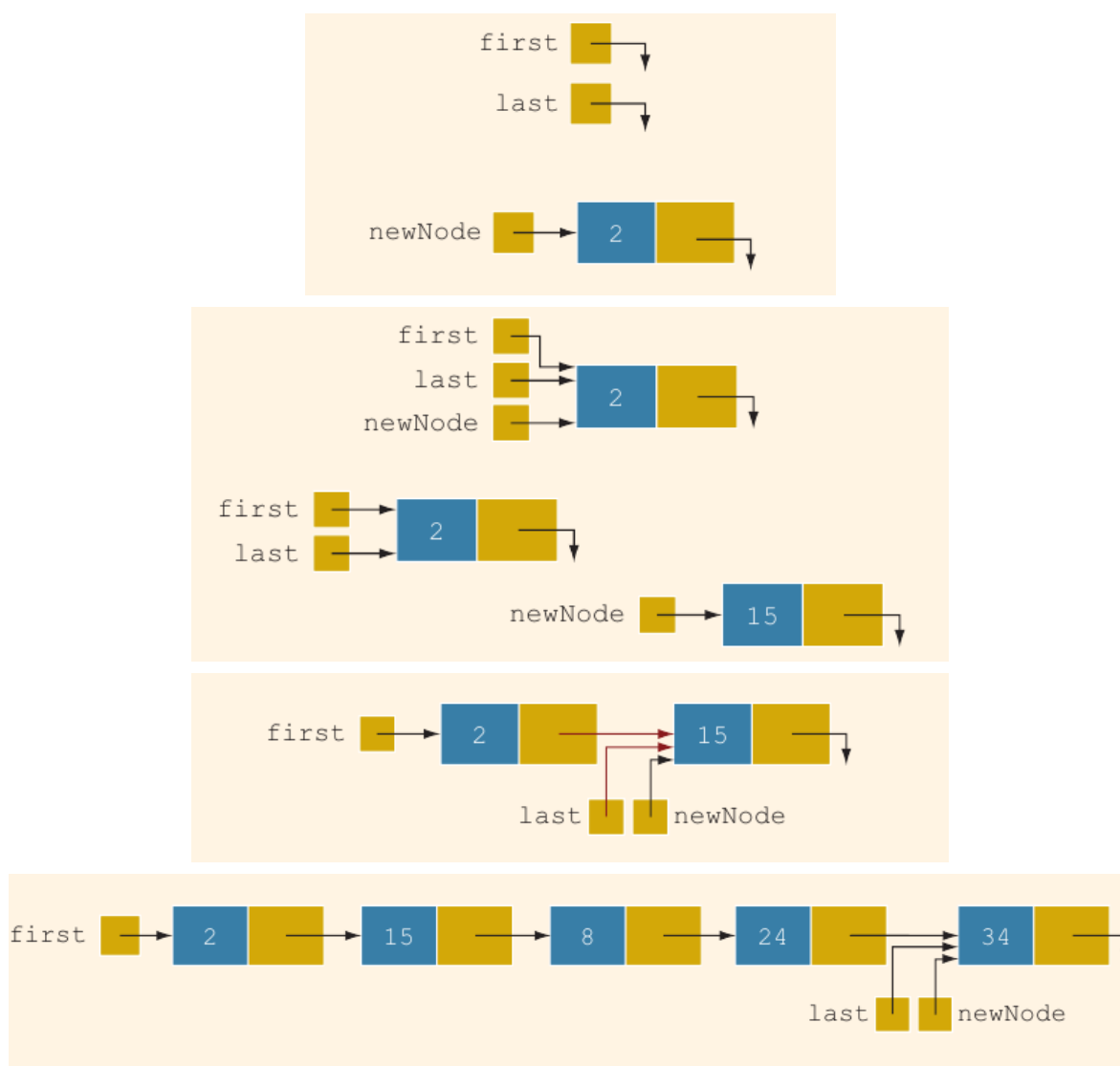
Чизиқли рўйхатни қуришда учта кўрсаткич керак бўлади.

```
nodeType *first, *last, *newNode;
int num;
```

Булар чизиқли рўйхатнинг бошини, охирини ва янги элементни кўрсатиб туриши учун керак.

```
first = NULL;
last = NULL;
cin >> num;
newNode = new nodeType;
newNode->info = num;
newNode->link = NULL;
if (first == NULL)
{
    first = newNode;
    last = newNode;
}
else
{
    last->link = newNode;
    last = newNode;
}
```

Аввал чизикли рўйхат боши ва охириги NULL қиймат берилди. Кейин рўйхатга қўшиш керак бўлган ўзгарувчи ўқиб олинди. Хотирада янги тугун яратилди ва унинг info майдонига экрандан киритилган сон юкланди. Янги тугуннинг кўрсаткичи NULLни кўрсатиб турибди. Сабаби, рўйхат охиридан тўлдирилмоқда, яъни янги элемент рўйхатнинг охирига жойлашади. Кейин текшириш амалга оширилган. Агар first NULL бўлса, яъни рўйхат ҳали яратилмаган бўлса, рўйхатнинг боши ҳам, охири ҳам янги тугунга айлантирилади. Акс ҳолда, рўйхатнинг охириги тугуни кўрсаткичи янги тугунни кўрсатади, сўнгра янги тугун рўйхатнинг охири сифатида last га юкланади.



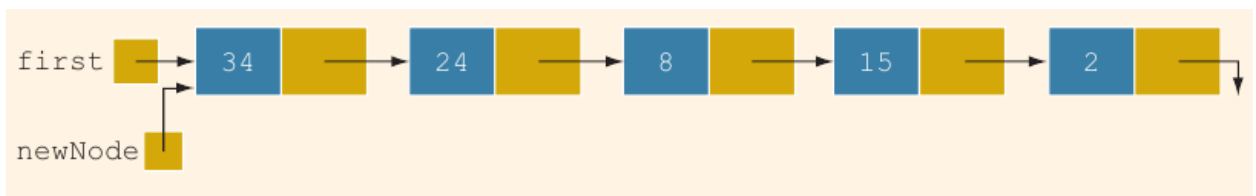
```
nodeType* buildListForward()
{
    nodeType *first, *newNode, *last;
    int num;
```

```

cout << "Sonni kiriting (tugash soni - 0): " << endl;
cin >> num;
first = NULL;
while (num != 0)
{
    newNode = new nodeType;
    newNode->info = num;
    newNode->link = NULL;
    if (first == NULL)
    {
        first = newNode;
        last = newNode;
    }
    else
    {
        last->link = newNode;
        last = newNode;
    }
    cin >> num;
}
return first;
}

```

Чизиқли рўйхатни бошидан куриб келиш ҳам мумкин. Бунда охириги тугунни эса сақлаб турувчи ўзгарувчидан фойдаланиш зарурати йўқолади.



```

nodeType* buildListBackward()
{
    nodeType *first, *newNode;
    int num;
    cout << "Sonni kiriting (tugash soni - 0): " << endl;
    cin >> num;
}

```



```

first = NULL;
while (num != 0)
{
    newNode = new nodeType;
    newNode->info = num;
    newNode->link = first;
    first = newNode;
    cin >> num;
}
return first;
}

```

### 29.3. Чизиқли рўйхат устида амаллар.

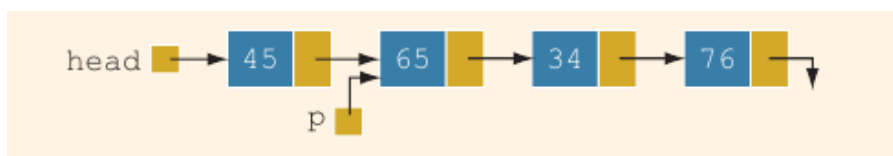
Чизиқли рўйхатга элемент қўшишда қўшимча кўрсаткич элементларидан фойдаланилади.

```

struct nodeType
{
    int info;
    nodeType *link;
};
nodeType *head, *p, *q, *newNode;

```

Элемент қўшишдан олдин чизиқли рўйхат куйидагича кўринишда бўлсин:



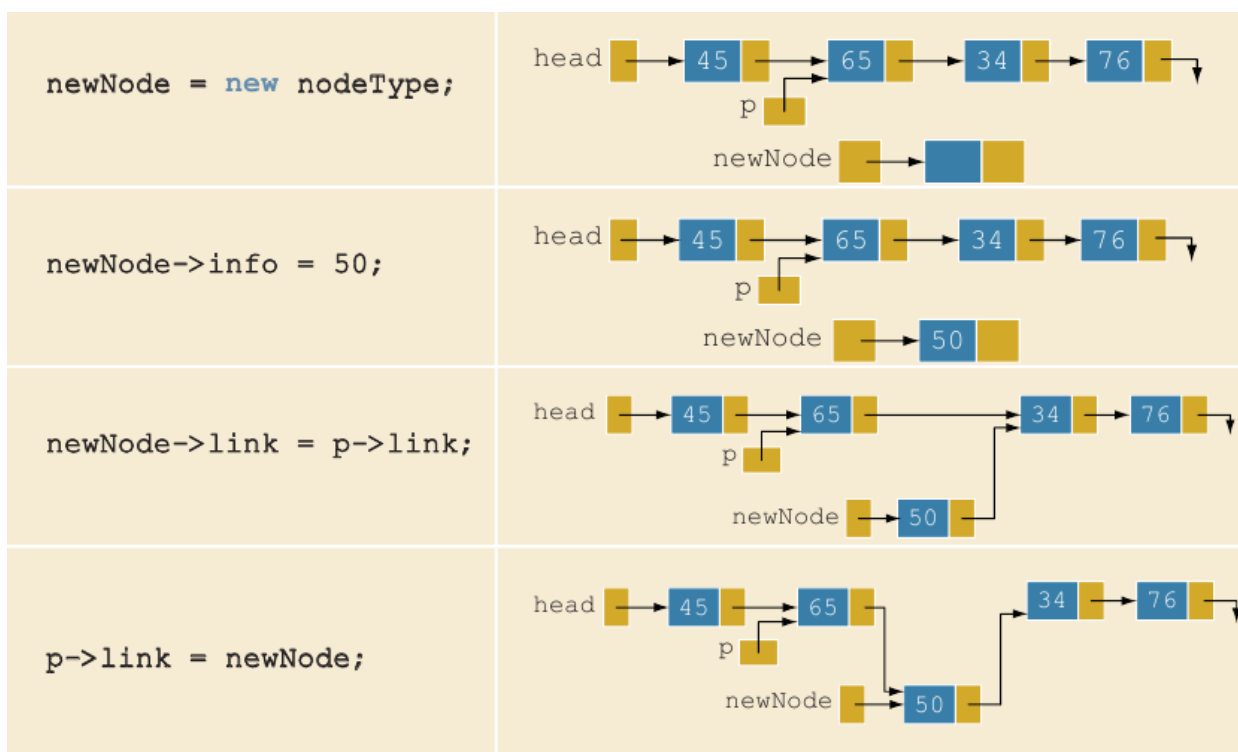
Янги элементни *p* кўрсаткичдан кейин қўшиш керак бўлса, куйидагича амаллар кетма-кетлиги бажарилади:

```

newNode = new nodeType; //янги тугун
newNode->info = 50; //янги тугун info майдони қиймати
newNode->link = p->link;
p->link = newNode;

```

Хар бир қатор бажарилишида хотирада элементлар куйидагича жойлашади:

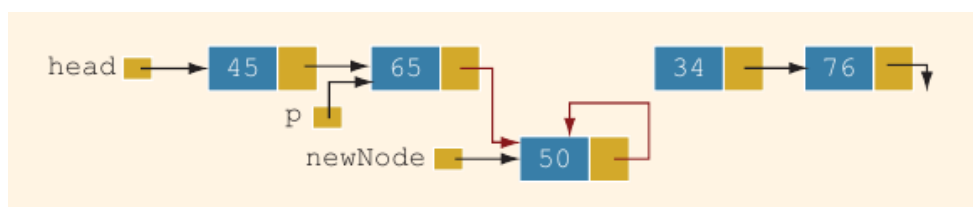


Дастур қисмининг айнан шу кўринишда ёзилиши муҳим аҳамиятга эга. Агар қаторлар ўрни алмашиб қолса, чизикли рўйхатнинг элементлари йўқолиб қолиши юзага келади. Масалан:

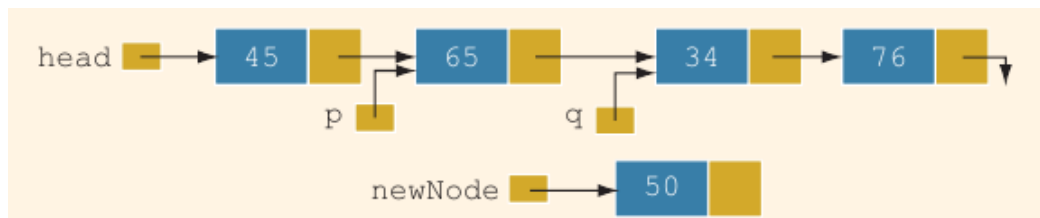
```

p->link = newNode;
newNode->link = p->link;

```



Ушбу ҳолатда, newNode тугуни кейинги тугун сифатида ўзини кўрсатиб турибди. Натижада охирига иккита тугунга боғланиш йўқолиб қолди. Дастур кодининг бундай кетма-кетликда ёзилишида сўнгги иккита тугунни йўқотиб кўймаслик учун қўшимча кўрсаткичдан фойдаланиш керак.



Бунда q кўрсаткичи учинчи тугунни кўрсатиб турибди. Ушбу ҳолатда янги тугунни қўшишда қандай кетма-кетликда ёзиш муҳим эмас, чунки q кўрсаткичи орқали охириги иккита тугунни ихтиёрий пайт қўшиб олиш мумкин:

```
newNode->link = q;
```

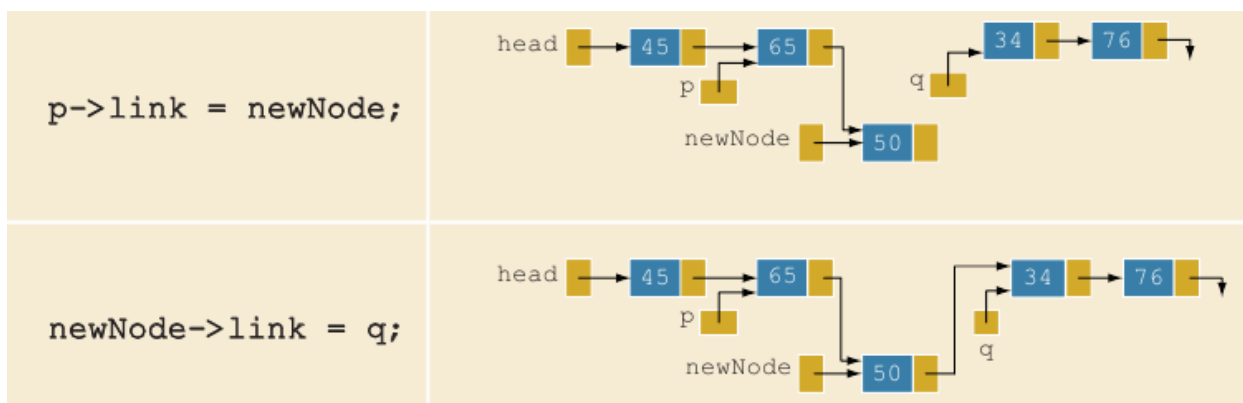
```
p->link = newNode;
```

Шунингдек, қуйидаги кодни ҳам ёзиш мумкин:

```
p->link = newNode;
```

```
newNode->link = q;
```

Хар бир қатор бажарилишида хотирада элементлар қуйидагича жойлашади:

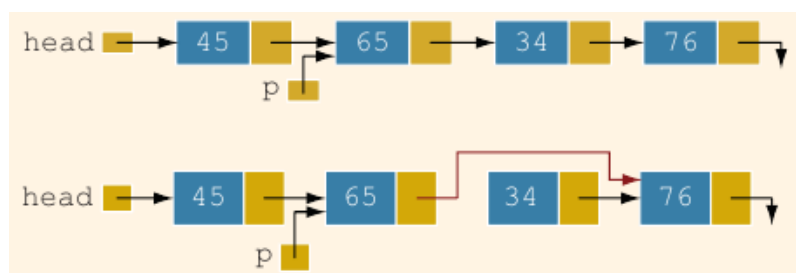


Бирор тугунни ўчириш керак бўлса икки хил усулдан фойдаланиш мумкин:

- 1) кўрсаткичнинг ўзи орқали
- 2) қўшимча кўрсаткич орқали.

Биринчи усулда кўрсаткичга кўрсаткич кўрсаткичини юклаш орқали амалга оширилади.

```
p->link = p->link->link;
```

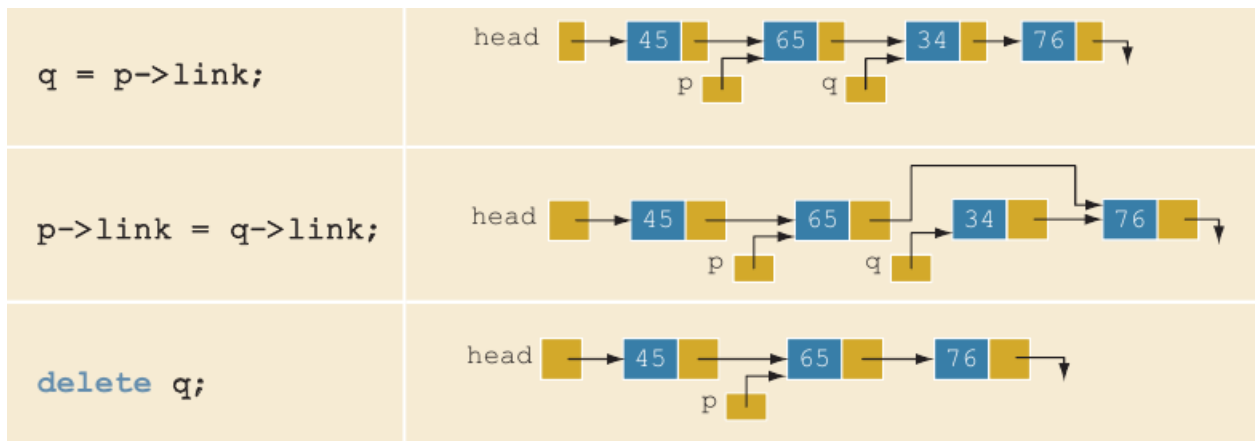


Иккинчи усулда қўшимча кўрсаткични қайси тугунгача ўчириш кераклигини кўрсатган ҳолда, бир бирига юклаш орқали ўчирилади.

```
q = p->link;
```

```
p->link = q->link;
```

```
delete q;
```



**Масала.** Нолдан фарқли бутун сонлардан иборат чизикли рўйхат яратилсин ва ундан кўрсатилган сонга тенг элемент ўчирилсин.

Бутун сонларнинг чизикли рўйхат кўринишидаги динамик структураси куйидаги майдонлардан ташкил топади:

```
struct Zanjir
{
    int element;
    Zanjir * keyingi;
};
```

Дастур матни:

```
#include <iostream>
using namespace std;
struct Zanjir
{
    int element;
    Zanjir * keyingi;
};
Zanjir * Element_Joylash(Zanjir * z, int yangi_elem);
{
    Zanjir * yangi=new Zanjir;
    yangi->element=yangi_elem;
    yangi->keyingi=0;
    if(z) // ro'yxat bo'sh emas
```

```

{
Zanjir * temp=z;
while(temp->keyingi)
temp=temp->keyingi;//ro'yxat oxirgi elementini topish
temp=temp->yangi;//elementni ro'yxat oxiriga qo'shish
}
else z=yangi; //ro'yxat bo'sh
return z; // ro'yxat boshi adresini qaytarish
}
Zanjir * Element_Uchirish(Zanjir * z,int del_elem);
{
if(z)
{
Zanjir * temp=z;
Zanjir * oldingi=0; // joriy elementdan oldingi
// elementga ko'rsatkich

while(temp)
{
if(temp->element==del_elem)
{
if(oldingi) //o'chiriladigan element birinchi emas
{
// o'chiriladigan elementdan oldingi elementni
// keyingi elementga ulash
oldingi->keyingi = temp->keyingi;
delete temp; //elementni o'chirish
temp=oldingi->keyingi;
}
else
{
//o'chiriladigan element ro'yxat boshida
z=z->keyingi;
delete temp;
temp=z;

```

```

}
}
else //element qiymati o'chiriladigan songa ten emas
{
oldingi=temp;
temp=temp->keyingi;
}
}
}
return z;
}
void Zanjir_Ekranga(Zanjir * z)
{
cout<<"Zanjir elementlari:"<<endl;
Zanjir * temp=z;
while(temp)
{
cout<<temp->element<<' ';
temp=temp->keyingi;
}
cout<<endl;
}
Zanjir * Zanjirni_Uchirish(Zanjir * z);
{
Zanjir * temp=z;
while(z)
{
z=z->keyingi;
delete temp;
}
return z;
}
int main()
{

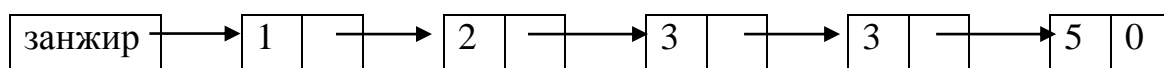
```

```

Zanjir * zanjir=0;
int son,del_element;
do
{
cout<<"\n Sonni kiriting (0-jarayonni tugatish): ";
cin>>son;
if(son)zanjir=Element_Joylash(zanjir,son);
}while(son);
Zanjir_Ekranga(zanjir);
Cout<<"\n O'chiriladigan elementni kiriting: ";
Cin>>del_element;
zanjir=Element_Uchirish(zanjir,del_element);
Zanjir_Ekranga(zanjir);
Zanjir = Zanjirni_Uchirish(zanjir);
return 0;
}

```

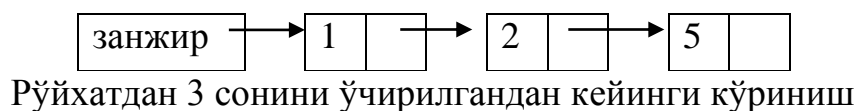
Программанинг бош функциясида чизикли рўйхат ҳосил қилиш учун Занжир туридаги занжир ўзгарувчиси аниқланган бўлиб, унга 0 қиймати берилган (бўш кўрсаткич қиймати, унинг эквиваленти - NULL). Кейин такрорлаш оператори танасида клавиатурадан бутун сон ўкилади ва Element\_Joylash() функциясини чақириш орқали бу сон рўйхатга охирига қўшилади. Функция янги ҳосил бўлган рўйхат бошининг адресини яна занжир ўзгарувчисига қайтаради. Агар клавиатурадан 0 сони киритилса рўйхатни ҳосил қилиш жараёни тугайди. Фараз қилайлик қуйидаги сонлар кетма-кетлиги киритилган бўлсин: 1,2,3,3,5,0. У ҳолда ҳосил бўлган рўйхат қуйидаги кўринишда бўлади:



Бешта сондан ташкил топган чизикли рўйхат

Ҳосил бўлган рўйхатни кўриш учун Zanjir\_Ekranga() функцияси чақирилади ва экранда рўйхат элементлари чоп этилади. Рўйхат устида амал сифатида берилган сон билан устма-уст тушадиган элементларни ўчириш

масаласи қаралган. Бунинг учун ўчириладиган сон дел\_элемент ўзгарувчига ўкилади ва у Element\_Uchirish() функцияси чақирилишида аргумент сифатида узатилади. Функция бу сон билан устма-уст тушадиган рўйхат элементларини ўчиради (агар бундай элемент мавжуд бўлса) ва ўзгарган рўйхат бошининг адресини занжир ўзгарувчисига қайтариб беради. Масалан, рўйхатдан 3 сони билан устма-уст тушадиган элементлар ўчирилгандан кейин у қуйидаги кўринишга эга бўлади:

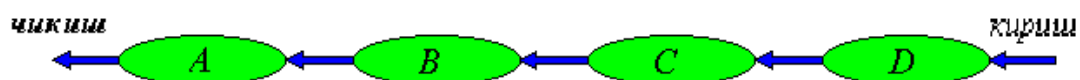


Ўзгарган рўхат элементлари экранга чоп этилади. Программа охирида, Zanjirni\_Uchirish() функциясини чақириш орқали рўйхат учун динамик равишда ажратилган хотира бўшатилади (гарчи бу ишнинг программа тугаши пайтида бажарилишининг маъноси йўк).

#### 29.4. Навбат

Кундалик ҳаётда деярли ҳар куни ҳар бир инсон навбат тушунчаси билан дуч келади. Умуман олганда навбат элементи қандайдир хизмат кўрсатишга буюртма бўлиб ҳисобланади: масалан, маълумотлар бюросидан керакли маълумотни олиш, кинотеатрларда чипта олиш, дўконда харид қилиб олинган маҳсулотларга кассада пул тўлаш ва бошқа.

Дастурлашда шундай маълумотлар тузилмаси мавжудки, у навбат дейилади. Бундай маълумотлар тузилмаси реал навбатни моделлаштиришда катта аҳамиятга эга. Бунда хизмат кўрсатишга келиб тушган талаб, унинг ижроси, яъни хизмат кўрсатиш тартибини аниқлашда зарур бўлади. Кундалик ҳаётимиздан барчамизга маълум бўлган навбат тури, дастурлашда FIFO (First Input – First Output, яъни биринчи келган - биринчи кетади) деб номланади. Қуйида 4 та элементдан иборат навбат келтирилган.



Бу ердан кўриниб турибдики, стекдан фарқли равишда хизмат кўрсатилиш биринчи келган элементга биринчи бўлиб хизмат кўрсатилади. Стекдан яна бир фарқи, бунда навбатнинг ҳар иккала томони очиқ бўлади, яъни бир томондан келиб иккинчи томондан чиқиб кетади.



Демак, навбатда элементни олиш рўйхат бошидан, ёзиш эса охиридан амалга оширилади.

ЕХМ хотирасида реал навбат элементлари сони чекли бўлган бир ўлчамли массив кўринишида яратилади. Албатта, бунда навбат элементи турини кўрсатиш ва навбат билан ишлашни кўрсатувчи ўзгарувчи зарур бўлади.

Навбат физик босқичда хотира соҳасини рўйхат кетма-кетлиги бўйича тўлалигича эгаллайди.

Навбат учун одатда 3 та оддий амал аниқланган.

1. Навбатга янги элемент жойлаштириш: `insert(q, x)`, бу ерда `q` - навбат, `x` - элемент.
2. Навбат бошидан элементни ўчириш: `remove(q)`
3. Навбатни бўш ёки бўш эмаслигини аниқлаш: `empty(q)`

Бундан ташқари, навбат бир ўлчамли массив кўринишида ифодаланганлиги учун массивни тўла ёки тўла эмаслигини кузатиб туриш лозим бўлади. Шу мақсадда, `full(q)` амалини ҳам киритиш мумкин.

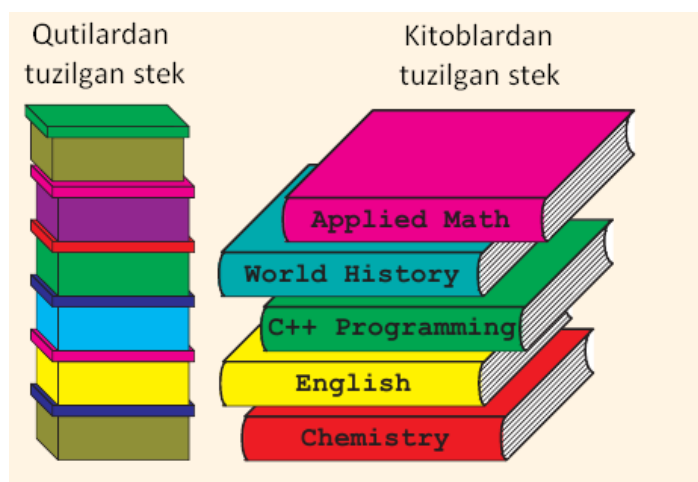
Умуман олганда, `insert` амалини ҳар доим бажариш мумкин. Сабаби, навбатни ташкил қилувчи элементлар сонига чекланишлар қўйилмаган. `remove` амали эса фақатгина навбат бўш бўлмагандагина ишлайди. `empty` амали эса ҳар доим ўринли.

## 29.5. Стек

LIFO (last in – first out), яъни навбатнинг охирги бўлиб кирган элементига биринчи бўлиб хизмат кўрсатилади. Бу энг кўп ишлатиладиган маълумотлар тузилмаларидан бири бўлиб, турли хил масалаларни ҳал қилишда анча қулай ва самарали ҳисобланади.

Хизмат кўрсатишни келтирилган тартибига кўра, стекда фақатгина битта позитсияга мурожаат қилиш мумкин. Бу позиция стекнинг учи дейилиб унда стекка вақт бўйича энг охирги келиб тушган элемент назарда тутилади. Биз стекга янги элемент киритсак, бу элемент олдинги стек учига турган элемент устига жойлаштирилади ҳамда стекни учига жойлашиб қолади. Элементни фақатгина стек учидан танлаш мумкин; бунда танланган элемент стекдан чиқариб ташланади ва стек учини эса чиқариб ташланган элементдан битта олдин келиб тушган элемент ташкил қилиб қолади. (бундай тузилмага маълумотларга чекланган мурожаат тузилмаси дейилади).

Стегни график кўринишида қуйидагича тасвирлаш мумкин:



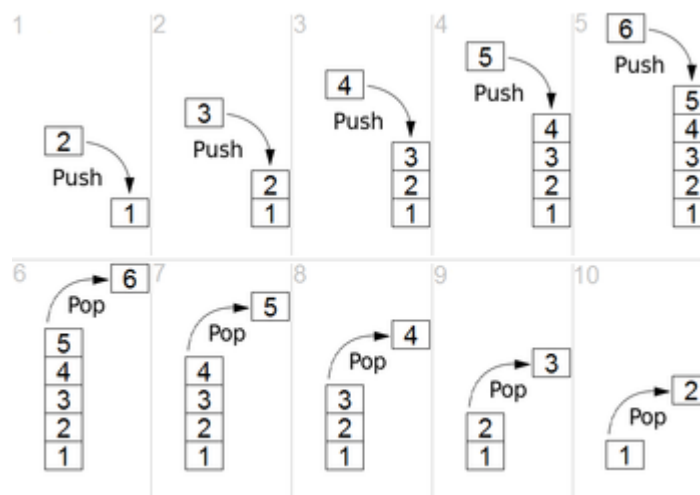
Структура орқали стекнинг кўриниши, унга элемент қўшиш ва ўчириш қуйидагича аниқланиши мумкин:

```
struct stack
{
    char *data;
    struct stack *next;
};
// Стегга янги элемент қўшиш функцияси
void push( STACK *ps, int x )
{
    if ( ps->size == STACKSIZE )
    {
        fputs( "Error: stack overflow\n", stderr );
        abort();
    }
    else
    {
        ps->items[ps->size++] = x;
    }
}
// Стегдан элемент ўчириш функцияси
int pop( STACK *ps )
```

```

{
    if ( ps->size == 0 )
    {
        fputs( "Error: stack underflow\n", stderr );
        abort();
    }
    else
    {
        return ps->items[--ps->size];
    }
}

```



#### Адабиётлар рўйхати:

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Мустақил таълим

1. Деклар ва улар билан ишлаш.
2. Массив орқали стек қуриш.
3. Массив орқали навбатни имитация қилиш.
4. Динамик тузилмалар билан ишловчи кутубхоналар.

## Глоссарий

Термин	Terminology	Ўзбек тилидаги шарҳи
include		препроцессор директиваси, кутубхона файлларни дастурга улаш учун ишлатилади
Кўрсаткич	indicator	қийматлари          адрес бўлган ўзгарувчилар
Адрес	adress	ўзгарувчи          хотирада жойлашадиган манзил
cout		экранга чиқариш оқими
cin		экрандан киритиш оқими
Структура	Struct	бир ёки ҳар хил турдаги берилганларни жамланмаси
Функция	Function	программинг алоҳида бўлаги, асосий қисм томонидан чақириб ишлатилади
Параметр	Parametr	функция ишлаши учун керак берилганлар
Ўзгарувчи	variable	берилганларни сақлаб туриш учун ишлатилувчи тил бирлиги

## Кейслар банки

**Кейс 1.** Дастур ишлаб чиқилди ва ушбу дастур талаба динамик тузилмасининг кўрсатилган ўрнига янги маълумотларни қўшиш ва уларни кўриш учун мўлжалланган. Дастур ишлаши жараёнида янги қўшилган маълумотдан кейин жойлашиши керак бўлган талабалар маълумотлари йўқолиб кетди. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб турли усулларда дастур кодини тақдим этинг (индивидуал ҳолда).

### Назорат саволлари

1. Берилганларнинг қандай динамик тузиилмалари мавжуд?
2. Чизикли рўйхат қандай ҳосил қилинади?
3. Стек қандай кўринишда ишлайди?
4. Навбат қандай кўринишда ишлайди?
5. Навбат устида қандай амаллар бажарилади?
6. Дек нима ва у қандай принципда ишлайди?
7. Стек кўринишига мисоллар келтиринг.
8. Чизикли рўйхат тугунлари қандай қилиб бир-бири билан боғланади?
9. Қандай ҳолларда чизикли рўйхат билан ишлаш хатолари юзага келади?
10. Стекка элемент жойлаш ва ўчириш амаллари қандай кўринишда бажарлади?

## Тест саволлари

1. Стекда элементларга хизмат кўрсатиш қандай кўринишда амалга оширилади?

- a. LIFO (last in – first out), яъни навбатнинг охириги бўлиб кирган элементига биринчи бўлиб хизмат кўрсатилади
- b. FIFO (first in – first out), яъни навбатнинг биринчи бўлиб кирган элементига биринчи бўлиб хизмат кўрсатилади
- c. FLIFO (first last in – first out), яъни навбатнинг биринчи ва охириги бўлиб кирган элементига биринчи бўлиб хизмат кўрсатилади
- d. CIFO (center in – first out), яъни навбат ўртасидаги элементига биринчи бўлиб хизмат кўрсатилади

2. Навбатда элементларга хизмат кўрсатиш қандай кўринишда амалга оширилади?

- a. LIFO (last in – first out), яъни навбатнинг охириги бўлиб кирган элементига биринчи бўлиб хизмат кўрсатилади
- b. FIFO (first in – first out), яъни навбатнинг биринчи бўлиб кирган элементига биринчи бўлиб хизмат кўрсатилади
- c. FLIFO (first last in – first out), яъни навбатнинг биринчи ва охириги бўлиб кирган элементига биринчи бўлиб хизмат кўрсатилади
- d. CIFO (center in – first out), яъни навбат ўртасидаги элементига биринчи бўлиб хизмат кўрсатилади

3. Чизиқли рўйхат нималардан ташкил топади?

- a. тугунлардан
- b. сатрлардан
- c. ўлчамлардан
- d. массивлардан

4. Рўйхатнинг охириги бўлиб кирган элементига биринчи бўлиб хизмат кўрсатиш нимада ишлатилади?

- a. стек
- b. навбат
- c. вектор
- d. матрица

5. Рўйхатнинг биринчи бўлиб кирган элементига биринчи бўлиб хизмат кўрсатиш нимада ишлатилади?

- a. стек
- b. навбат
- c. вектор
- d. матрица

6. Стекка элемент жойлаш учун қайси функция ишлатилади?

- a. pop
- b. push
- c. empty
- d. free

7. Стекдан элемент олиш учун қайси функция ишлатилади?

- a. pop
- b. push
- c. empty
- d. free

8. Қуйидаги дастур кодининг вазифаси нимадан иборат?

```
current = head;
while (current != NULL)
{
    current = current->link;
}
```

- a. чизиқли рўйхат элементларида ҳаракатланиш
- b. чизиқли рўйхат элементларини ўчириш
- c. чизиқли рўйхатга элемент қўшиш
- d. чизиқли рўйхатдан элемент излаш

9. Қуйидаги дастур кодининг вазифаси нимадан иборат?

```
struct Vaqt{unsigned int h,m,s;};
int main(int argc, char* argv[])
{
```

```

Vaqt a={0,59,59},b={0,5,5};    a.m=b.h+b.s;
cout<<a.h<<":"<<a.m<<":"<<a.s<<endl;
return 0;
}

```

- a. 0:5:59
- b. 0:6:59
- c. 0:5:61
- d. 0:5:5

10. Қуйидаги дастур кодининг вазифаси нимадан иборат?

```

struct Racional{int surat,maxraj;};
void main(int argc, char* argv[])
{
Racional a={52,33}, b={12,4};
b.surat=(a.surat+4)/a.maxraj;
a.surat=(a.surat+b.maxraj)/b.surat;
cout<<b.surat<<"/"<<a.maxraj<<endl;
}

```

- a. 1/33
- b. 5/33
- c. 52/33
- d. 12/4



## V. АМАЛИЙ МАШҒУЛОТ МАТЕРИАЛЛАРИ

### Амалий машғулот 24-25.

#### Функция ва массивлар

**Ишдан мақсад:** C++ дастурлаш тилида функциялар ва массивларни дастурда биргаликда ишлаш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда статик ва динамик массивлардан функция параметри, қайтарувчи қиймат тури сифатида фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

#### Ишни бажариш учун намуна

**Мисол:** Берилган  $n \times m$  ўлчамдаги матрисанинг  $k$ -устундаги энг катта элементи топилсин ( $0 \leq k \leq m$ ).

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;

int K_ustun_max(int n,int k,int **a)
{
    int max=a[0][k];
    for(int i=1;i<n;i++)
        if(max<=a[i][k]) max=a[i][k];
    return max;
}

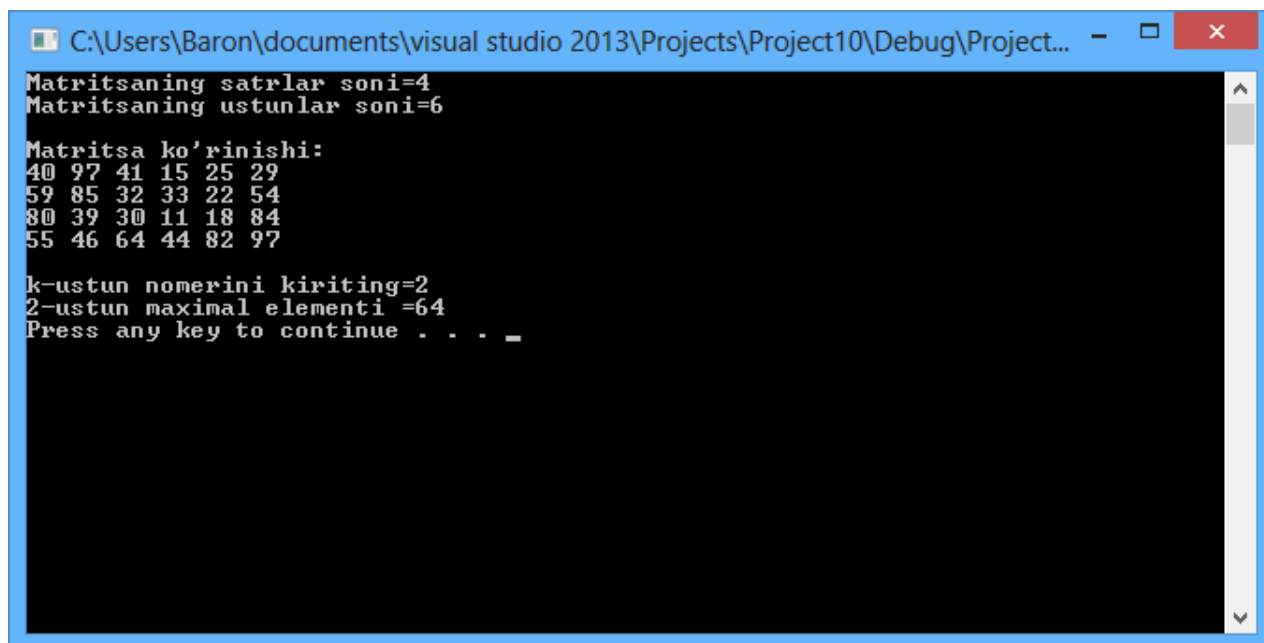
int main()
{
    int n, m, k, l;
    int **a;
    cout << "Matritsaning satrlar soni=";
```

```

cin >> n;
cout << "Matritsaning ustunlar soni=";
cin >> m;
// matritsani hosil qilish
a = new int *[n];
for (int i = 0; i < n; i++)
    a[i] = new int[m];
srand(time(0));
// matritsa elementlari qiymatlarini tanlab olish
for (int i = 0; i < n; i++)
for (int j = 0; j < m; j++)
    a[i][j] = rand() % 90 + 10;
// matritsani chop etish
cout << "\nMatritsa ko'rinishi:\n";
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < m; j++)
        cout << a[i][j] << " ";
    cout << endl;
}
do
{
    cout << "\nk-ustun nomerini kiriting="; cin >> k;
} while (k < 0 || k >= m);
cout << k << "-ustun maximal elementi =" <<
K_ustun_max(n, k, a);
for (int i = 0; i < n; i++)
    delete[]a[i];
delete[]a;
system("pause");
return 0;
}

```

Дастур ишлаши натижаси:



```
C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project... - □ ×
Matritsaning satrlar soni=4
Matritsaning ustunlar soni=6

Matritsa ko'rinishi:
40 97 41 15 25 29
59 85 32 33 22 54
80 39 30 11 18 84
55 46 64 44 82 97

k-ustun nomerini kiriting=2
2-ustun maximal elementi =64
Press any key to continue . . . _
```

### Амалий топшириқлар

1. Ҳар бири 30 та бутун сондан иборат иккита кетма-кетлик берилган. Биринчи кетма-кетликнинг иккинчи кетма-кетликка кирмаган сонлари ичидан энг кичиги топилсин (бунақа сонлардан камида биттаси мавжуд деб фараз қилинсин).

2. `const int n=10;`  
`float D[n][n],S;`

Берилган D матритса учун  $S = \sum_{k=1}^n \max_{1 \leq i, j \leq k} D_{ij}$  ҳисоблансин.

3. `int A[15][20], b[15];`

Берилган A матритсадан b вектор ҳосил қилинсин. Қуйидаги кўрсатилган шартлар бажарилса, b[k] элементи true, акс ҳолда false қиймат қабул қилсин:

- а) A матритсанинг k-сатри ноллардан иборат;
- б) A матритсанинг k-сатр элементлари камайиш бўйича тартибланган;
- д) A матритсанинг k-сатри симметрик.

4. Коеффисентлари ҳақиқий турда бўлган n – тартибли “учбурчак” кўринишидаги чизиқли тенгламалар системаси ечилсин.

$$\left. \begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\dots \\ a_{nn}x_n &= b_n \end{aligned} \right\}$$

5. const n=4,m=5;

```
int A[n][m], B[n][m], C[n,m], N[n][m], D[2*n][3*m];
```

Берилган A, B ва C матритсалар орқали D матритсани курадиган

constr(A,B,C,D) функция тузилсин. D матритса  $D = \begin{pmatrix} A & B & C \\ B & N & A \end{pmatrix}$  кўринишида

аниқланади. Бунда N нол матритса.

6. Учта ҳақиқий турдаги 4-тартибли квадрат матритсалар берилган. Уларнинг орасида нормаси энг кичик бўлгани чоп этилсин (бундай матритса битта деб ҳисоблансин). Матритсанинг нормаси сифатида унинг элементлари абсолют қийматларининг максимуми олинсин.

7. Натурал p сони ва 4-тартибли ҳақиқий турдаги A, B ва C квадрат матритсалар берилган.  $(ABC)^p$  ҳосил қилинсин

8. Ҳақиқий турдаги 10x20 ўлчамли A, B ва C матритсалар берилган. Куйидаги катталиқ ҳисоблансин:

$$\frac{\|A\| + \|B\| + \|C\|}{\|A + B + C\|}, \text{ бу ерда } \|D\| = \max_j |D_{1,j}| + \max_j |D_{2,j}| + \dots + \max_j |D_{10,j}|.$$

9. Иккита 10-тартибли бутун турдаги квадрат матритса берилган. Матритса диагоналлариға нисбатан бир марта акслантириш орқали биридан иккинчисини ҳосил қилиш мумкинми?

10. Тасодифий сонларни ҳосил қилувчи функциядан фойдаланган ҳолда n та ўзаро тенг бўлмаган бутун сонлар массивини натижа сифатида қайтарадиган функция тузилсин.

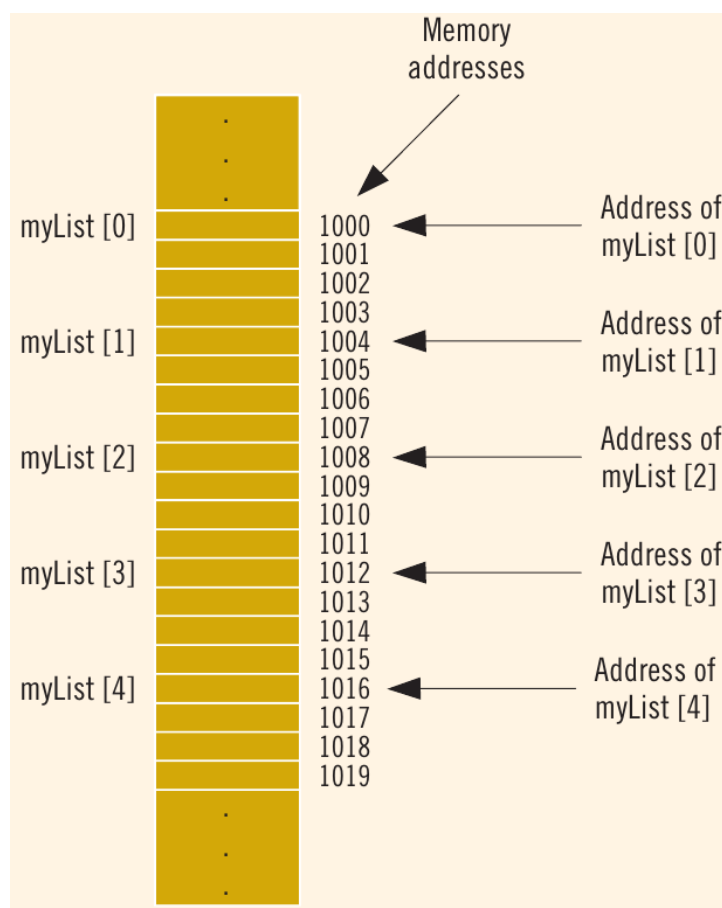
11. Тасодифий сонларни ҳосил қилувчи функциядан фойдаланган ҳолда бутун турдаги n-тартибли квадрат матритса ҳосил қилинсин (n=10). Матритсанинг энг катта ва энг кичик элементларини алмаштирувчи функция тузилсин.

## Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

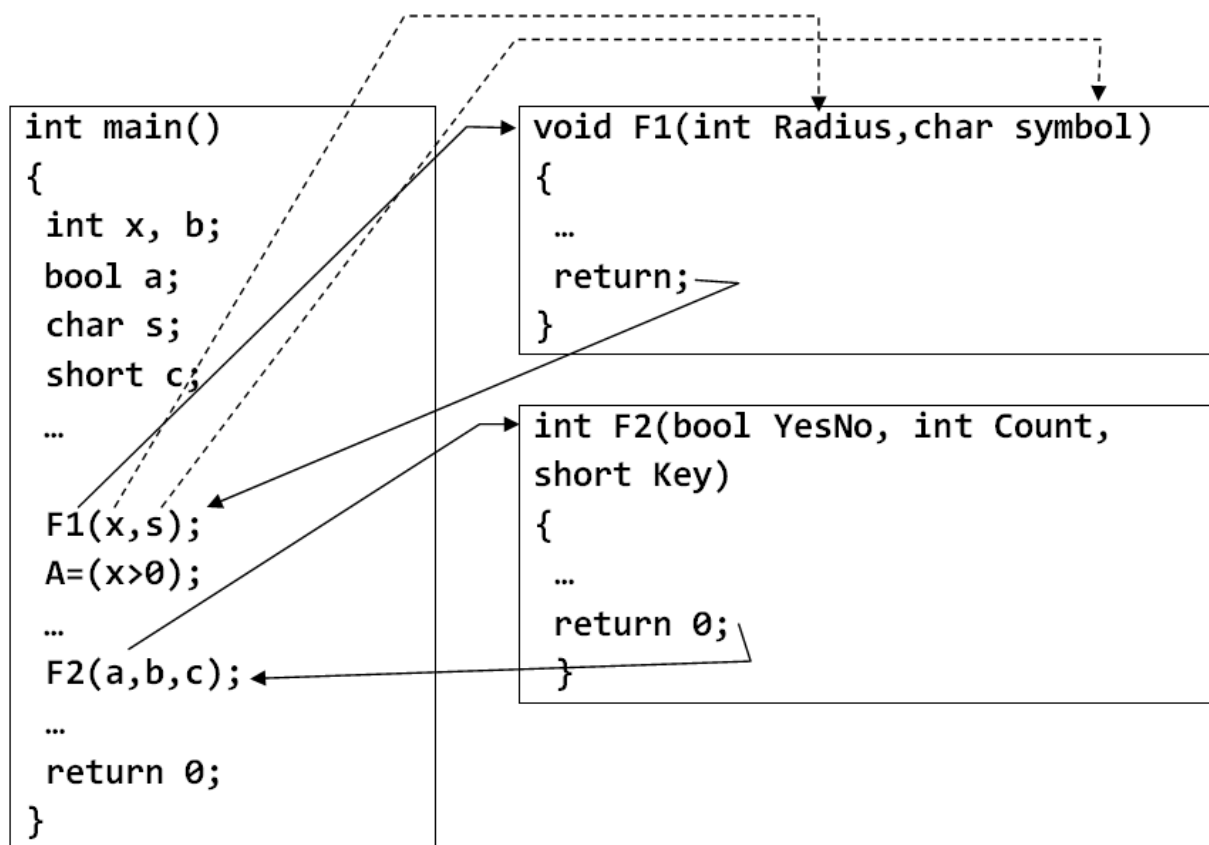
## Тарқатма материаллар.

1. `int myList[5];` нинг хотирада жойлашиши



Динамик хотирадан фой ажратиш

## 2. Функция ишлатилиши.



### Кейслар банки

**Кейс 1.** Дастур матрица қабул қилувчи ва вектор қайтарувчи функцияга эга. Функциянинг вазифаси матрица қаторларидаги минимум элементлардан вектор тузиш ва уни қайтаришдан иборат. Дастур ишлаши натижасида минимум элементлар нотўғри ҳисобланмоқда. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Дастурни тўғри ишлаши учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).
- C++ дастурлаш тили имкониятларидан келиб чиқиб бир нечта усулда дастур кодини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(double *a, int n)
{
for (int i=0;i<2;i++) a[i]=n*a[i+1];
return n;
}
void main()
{
double b[]={7,1,3}, a[]={0,0,3};
b[0]=F(b,3);
b[1]=F(a,5);
cout << b[0]-a[1]/b[2];
}
```

- a) -2
- b) 15
- c) 5
- d) 11

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(double *a, int n)
{
for (int i=0;i<2;i++) a[i]=n*a[i+1];
return n;
}
void main()
{
double b[]={7,1,3}, a[]={0,0,3};
b[0]=F(b,3);
a[1]=F(a,5);
cout << b[0]-a[0]/b[2];
}
```

- a) -2
- b) 3
- c) 5
- d) 4

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int n, int k, int t)
{
int i=5;
for (i=0; i!=0; )
    i--;
return n+i-k+t;
}
void main()
{
int a[]={2,5,2,6,7}, n=2, k=1, t=3;
for (int i=0;i<3;i++)
if (2*F(a[i], a[i], a[0])>t) t=t+F(a[i],t,a[0]);
cout << t;
}
```

- a) 4
- b) 5
- c) 6
- d) 7

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=4;
int a[n][n];
for(int i=0;i<n;i++)
    for(int j=0;j<n;j++) a[i][j]=1;
int s=0;
for(int i=0;i<n;i++) s+=a[i][i]+a[i][n-1-i];
cout<<s;
```



- a) 8
- b) 9
- c) 7
- d) 6

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(int n, int k=5)
{
k=n-k;
return n-k;
}
void main()
{
int n=7, a[]={6,2,3,8,1}, c=5, k=2;
for (int i=0;i<5;i++)
if (F(c,a[i]-n)-F(n-a[i],c)<0) c=c+F(c, c);
cout << c;
}
```

- a) 160
- b) 310
- c) 210
- d) 110

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a[]={15,30,25,10,45,20};
int s=0;
for(int i=1;i<5;i++) if(a[i]>a[i-1]&&a[i]>a[i+1])
s+=a[i];
cout<<s;
```

- a) 75
- b) 125
- c) 110
- d) 85

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int* F(double *a)
{
a[0]=11+a[1];
a[1]=12-a[0];
a[2]=a[1]+a[2];
}
void main()
{
double b[]={1,4,7};
F(b);
cout << b[0]+b[1]+b[2];
}
```

- a) 37
- b) 16
- c) 11
- d) 27

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
const int n=4;
int a[n][n], s=0;
for(int i=0;i<n;i++)
    {a[i][0]=1;
    for(int j=1;j<n;j++) a[i][j]=2*a[i][j-1];
    }
for(int i=0;i<n;i++) s+=a[i][i];
cout<<s;
```

- a) 15
- b) 25
- c) 20
- d) 18

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int F(double *a, int n)
{
for (int i=0;i<2;i++) a[i]=n*a[i+1];
return n;
}
void main()
{
double b[]={1,4,7};
b[2]=F(b, 6);
cout << b[0]-b[1]/b[2];
}
```

- a) 91
- b) 18
- c) 13
- d) 17

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int* F(int *a, int n=2)
{
if (!n) return a;
int *c=new int [3];
c[0]=3;
for (int i=0;i<2;i++) {c[i+1]=n*a[i+1];a[i+1]=c[i]-3;}
return c;
}
void main()
{
int b[]={7,1,3}, a[]={3,3,3}, *c;
c=F(b);
cout << c[1]*a[0]-a[1]/b[2]*c[3];
}
```

- a) 3
- b) 6
- c) 5
- d) 13

## Амалий машғулот 26-27. Тузилмалар. Бирлашмалар

**Ишдан мақсад:** C++ дастурлаш тилида тузилмалар, бирлашмалар билан ишлаш, тузилмалар, тузилмалар массиви устида турли амаллар бажариш, турли кўринишдаги тузилмалардан фойдаланиш кўникмаларига эга бўлиш. Дастурда турли кўринишдаги тузилмалар, бирлашмалардан фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Текисликда берилган  $n$  та  $p_i(x_i, y_i)$ ,  $i = \overline{1, n}$  нукталарни ўз ичига оладиган минимал радиусли айлана аниқлансин.

Ечиш усули: Барча нукталарни ўз ичига оладиган айлана маркази  $M(x_m, y_m)$  нукталарнинг геометрик маркази бўлади, бу ерда

$$x_m = \frac{\sum_{i=1}^n x_i}{n}, y_m = \frac{\sum_{i=1}^n y_i}{n} .$$
 Айлана радиуси сифатида айлана марказидан

нукталаргача бўлган масофаларнинг энг каттаси олинади -

$$r = \max_{1 \leq i \leq n} \left\{ \sqrt{(x_m - x_i)^2 + (y_m - y_i)^2} \right\}.$$

Дастур коди:

dastur.cpp файли:

```
#define _USE_MATH_DEFINES
#include <iostream>
#include <cmath>
using namespace std;

struct Nuqta
{
    float x, y;
};
```

```

Nuqta Aylana_Markazi(Nuqta*nuqta, int n)
{
    Nuqta N;
    N.x = 0; N.y = 0;
    for (int i = 0; i<n; i++)
    {
        N.x += nuqta[i].x; N.y += nuqta[i].y;
    }
    N.x /= n; N.y /= n;
    return N;
}
float Aylana_radiusi(Nuqta m, Nuqta*nuqta, int n)
{
    float r = 0, d;
    for (int i = 0; i<n; i++)
    {
        d = sqrt(pow(nuqta[i].x - m.x, 2) + pow(nuqta[i].y -
m.y, 2));
        if (d>r)r = d;
    }
    return d;
}
int main()
{
    int n;
    Nuqta a_markaz;
    float a_radius;
    cout << "Nuqtalar sonini kiriting: "; cin >> n;
    Nuqta *nuqta = new Nuqta[n];
    for (int i = 0; i<n; i++)
    {
        cout << "\nx[" << i + 1 << "]=";
        cin >> nuqta[i].x;
        cout << "y[" << i + 1 << "]=";
    }
}

```

```

        cin >> nuqta[i].y;
    }
    a_markaz = Aylana_Markazi(nuqta, n);
    a_radius = Aylana_radiusi(a_markaz, nuqta, n);
    cout << "\nAylana M(" << a_markaz.x << ', ' << a_markaz.y
<< "), ";
    cout << "R = " <<a_radius;
    system("pause");
    return 0;
}

```

Дастур ишлаши натижаси:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
Nuqtalar sonini kiriting: 4
x[1]=4
y[1]=3
x[2]=-2
y[2]=3
x[3]=2
y[3]=2
x[4]=5
y[4]=0
Aylana M(2.25,2), R = 3.40037
Press any key to continue . . .

```

### Амалий топшириқлар

1. `struct Domino_tosh{unsigned char chap,ung;};`  
`Domino_tosh domino[28];`  
 Берилган домино қатордаги домино тошлари тўғри қўйилганлигини (навбатдаги домино тошининг ўнг қисмидаги сон кейинги тошнинг чап қисмидаги сонга тенглигини) текширадиган `Tugri_qator(domino)` мантиқий функция тузилсин.

2. `struct Vaqt{unsigned char soat,min,sek;};`  
`Vaqt t1,t2;`  
 Қуйидаги масалалар ечилсин:
- $t_1$  вақт  $t_2$  вақтдан олдин келишини текширадиган олдин ( $t_1, t_2$ ) мантиқий функция тузилсин (бир сутка ичида);
  - $t$  вақтдан 1 секунд ортиқ вақтни  $t_1$  параметрга берувчи `secund(t,t1)` функция тузилсин (сутка алмашинуви ҳисобга олинган ҳолда);
  - $t_1$  вақтдан  $t_2$  вақтгача қанча вақт ўтганлигини ҳисоблайдиган `interval(d,t2,t1)` функцияси тузилсин ( $t_2 > t_1$ ).
3. `struct Rasional{int surat, maxraj;};`  
`Rasional r[20],a,b;`  
 Қуйидаги масалалар ечилсин:
- $a$  ва  $b$  рационал сонларнинг тенглигини текширувчи `teng(a,b)` мантиқий функцияси тузилсин;
  - $a$  ва  $b$  рационал сонлар йиғиндисини  $a$  ўзгарувчисига қайтарувчи `summa(a,b)` функцияси тузилсин;
  - $a$  рационал сонни қисқартириб бўлмайдиган кўринишга келтирадиган `qisqartir(a)` функцияси тузилсин;
  - $r$  рационал сонларнинг энг каттасини  $m$  параметрга берувчи `max(x,m)` функцияси тузилсин.
4. Ўқиш файлида талабаларнинг сессия натижалари ҳақидаги маълумот қуйидаги кўринишга эга:
- `<фамилия>,<гуруҳ номери>,<рейтинг1>,<рейтинг2>,<рейтинг3>`.
- Бу ерда фамилия 15 ҳарфгача бўлган сатр, гуруҳ номери - бутун сон, ҳар бир рейтинг 0 ва 100 оралигидаги ҳақиқий сон: рейтинг1- математик анализдан, рейтинг2 - алгебрадан, рейтинг3- программалаш предметларидан.
- Қуйидагилар аниқлансин:
- камида битта фандан қарздор бўлган талабанинг фамилияси;
  - барча имтиҳонларни 4 ва 5 баҳога топширган талабалар неча фоизни ташкил қилиши;
  - талабалар қайси фандан имтиҳонларни энг яхши натижа билан топширганлиги;
  - талабалар ўзлаштиришининг ўртача қиймати бўйича ўсмайдиган тартибдаги гуруҳларнинг тартиб номерлари.

5. АЕРОФЛОТ структураси <рейснинг манзил пункти>,<рейс номери> ва <самолет тури> майдонларини ўз ичига олади. Берилган н ўлчамидаги АЕРОФЛОТ туридаги массив яратилиб, маълумотлар киритилсин ва қуйидаги амаллар бажарилсин:
  - a) массив, рейс номерининг ўсиши бўйича тартиблансин;
  - b) клавиатурадан киритилган манзилга учадиган рейс номери ва самолет тури чоп этилсин;
  - d) клавиатурадан киритилган самолет тури учадиган рейс номерлари ва манзиллари чоп этилсин.
6. ИШЧИ структураси <фамилия ва инитсиаллари>,<лавозими> ва <ишга кирган йили> майдонларини ўз ичига олади. Берилган н ўлчамидаги ИШЧИ туридаги массив яратилсин ва қуйидаги амаллар бажарилсин:
  - a) массив ишчи фамилиясининг алфавит бўйича жойлашувига мос тартиблансин;
  - b) клавиатурадан киритилган йилдан кейин ишга кирган ишчилар фамилияси чоп этилсин;
  - d) клавиатурадан киритилган лавозимдаги ишчилар фамилияси ва ишга кирган йили чоп этилсин.
7. ПОЕЗД структураси <поезд номери>,<борадиган манзил номи> ва <жўнаш вақти> майдонларини ўз ичига олади. Берилган н ўлчамидаги ПОЕЗД туридаги массив яратилсин ва қуйидаги амаллар бажарилсин:
  - a) массив поезд номерларининг ўсиши бўйича тартиблансин;
  - b) клавиатурадан киритилган номерига мос поезд номери ҳақидаги маълумот чоп этилсин;
  - d) клавиатурадан киритилган манзилга ва кўрсатилган вақтдан кейин жўнайдиган поездлар ҳақидаги маълумот чоп этилсин.
8. МАРШРУТ структураси <маршрут боши>,<маршрут охири> ва <маршрут номери> майдонларини ўз ичига олади. Берилган н ўлчамидаги МАРШРУТ туридаги массив яратилсин ва қуйидаги амаллар бажарилсин:
  - a) массив маршрут номерларининг ўсиши бўйича тартиблансин;
  - b) клавиатурадан номери киритилган маршрут ҳақидаги маълумот чоп этилсин;
  - d) клавиатурадан киритилган ном бўйича, бошланаши ёки охири шу номдаги пункт бўлган маршрутлар ҳақидаги маълумот чоп этилсин.



9. БЛОКНОТ структураси шахс ҳақидаги - <фамилия ва исми>,<телефон номери> ва <туғилган санаси> майдонларини ўз ичига олади. Берилган п ўлчамидаги БЛОКНОТ туридаги массив яратилсин ва қуйидаги амаллар бажарилсин:
- a) массив одамнинг туғилган санасининг ўсиши бўйича тартиблансин;
  - b) клавиатурадан телефон номери киритилган одам ҳақидаги маълумот чоп этилсин;
  - d) туғилган оиди клавиатурадан киритилган сонга мос келувчи одамлар ҳақидаги маълумот чоп этилсин.
10. НАРХ структураси маҳсулот ҳақидаги <маҳсулот номи>,<маҳсулот сотиладиган магазин номи> ва <маҳсулотнинг сўмдаги нархи> майдонларини ўз ичига олади. Берилган п ўлчамидаги НАРХ туридаги массив яратилсин ва қуйидаги амаллар бажарилсин:
- a) массив маҳсулот номини алфавит бўйича жойлашувига мос тартиблансин;
  - b) номи клавиатурадан киритилган маҳсулот ҳақидаги маълумот чоп этилсин;
  - d) клавиатурадан киритилган номдаги магазинда сотиладиган маҳсулотлар рўхати ва уларнинг жами нархи чоп этилсин.
11. ОРДЕР структураси банк миждози ҳақидаги <тўловчининг ҳисоб рақами>,<олувчининг ҳисоб рақами> ва <ўтказиладиган пул миқдори> майдонларини ўз ичига олади. Берилган п ўлчамидаги ОРДЕР туридаги массив яратилсин ва қуйидаги амаллар бажарилсин:
- a) массив тўловчининг ҳисоб рақами бўйича тартиблансин;
  - b) ҳисоб рақами клавиатурадан киритилган тўловчининг ҳисоб рақамидан қанча пул олинганлиги ҳақидаги маълумотлар чоп этилсин;
  - d) ҳисоб рақами клавиатурадан киритилган олувчининг ҳисоб рақамига қайси ҳисобдан қанча пул келиб тушганлиги ҳақидаги маълумот чоп этилсин.

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

### 1. Структура кўриниши

```
struct <struktura nomi>
{
    <tur1 > <nom1>;
    <tur2 > <nom2>;
    ...
    <turn> <nomn>;
};
```

### 2. Структура майдонлари хажмини алоҳида кўрсатиш

```
struct Sana_vaqt2
{
    unsigned Yil:7;
    unsigned Oy:4;
    unsigned Kun:5;
    unsigned Soat:6;
    unsigned Minut:6;
    unsigned Sekund:6;
};
```

## Кейслар банки

**Кейс 1.** Тузилмалар билан ишлайдиган дастур тузилди. Дастур вазифаси китоб тузилмаси орқали муаллифнинг китоблари сони бўйича муаллифлар фамилия исmlарини ўсиш бўйича тартиблаб чиқаришдан иборат. Дастур ишга туширилганда тартиблаш жараёни бажарилмаётгани маълум бўлди. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тилида дастур кодани тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Uchburchak{int a,b,c};
void main(int argc, char* argv[])
{
Uchburchak a={4,5,6}, b={6,8,10};
int yuza=a.a+a.b+a.c;
int perimetr=b.a+b.b+b.c;
int balandlik=a.a/b.b-b.c;
Uchburchak c={yuza,perimetr,balandlik};
cout<<c.a<<"-"<<balandlik<<"-"<<b.b<<endl;
}
```

- a) 15--10-8
- b) 15-10-8
- c) 5-10-8
- d) 15--10-10

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Mashina{char Nomi[20]; unsigned int Tezligi,
Narxi};
int main(int argc, char* argv[])
{
Mashina
m[]={{"Ford",250,60000}, {"Toyota",300,70000}, {"Ferrari",
350,90000}};
cout<<(m+1)->Nomi<<" , " <<(m+1)->Tezligi<<" , " <<(m+1)-
>Narxi<<endl;
return 0;
}
```

- a) Toyota, 300 , 70000
- b) Ford, 250 , 70000
- c) Toyota, 300 , 60000
- d) Ferrari, 350, 90000

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Racional{int surat,maxraj};  
void main(int argc, char* argv[])  
{  
Racional a={52,12}, b=a;  
a.maxraj=13;  
b.surat=(a.surat+4)/a.maxraj+a.surat;  
a.surat=(a.surat+b.maxraj)/b.surat;  
a=b;  
cout<<b.surat<<"/"<<a.surat<<endl;  
}
```

- a) 56/56
- b) 1
- c) 56/12
- d) 12/56

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Vaqt{unsigned int soat,minut,sekund};  
void main()  
{  
Vaqt a={0,13,59},b={3,52,5}; a.sekund=a.minut+59;  
a.minut=a.sekund+32;  
cout<<a.soat<<":"<<a.minut<<":"<<a.sekund<<endl;  
}
```

- a) 0:104:72
- b) 1:54:72
- c) 1:55:12
- d) 0:105:12

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Uchburchak{int a,b,c;};  
void main(int argc, char* argv[])  
{  
Uchburchak b={1,2,2}, a={33,3,7},c=b;  
int yuza=a.a*a.b*c.c;  
int yza=a.a+b.b+a.c;  
cout<<yuza;  
}
```

- a) 198
- b) 212
- c) 205
- d) 172

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Vaqt{unsigned int soat, minut, sekund;};  
int main(int argc, char* argv[])  
{  
Vaqt a={2,34,11},b={1,12,5}; b.minut=a.minut+9;  
a.sekund=b.sekund+a.minut;  
cout<<b.minut<<":"<<a.sekund<<":"<<b.soat<<endl;  
return 0;  
}
```

- a) 43:39:1
- b) 50:39:1
- c) 43:42:1
- d) 43:39:14

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Racional{int surat,maxraj;};  
void main(int argc, char* argv[])  
{  
Racional a={52,33}, b=a;
```

```

a.maxraj=13;
b.surat=(a.surat+4)/a.maxraj-a.surat;
a.surat=(a.surat+b.maxraj)/b.surat;
cout<<b.surat<<"/"<<a.maxraj<<endl;
}

```

- a) -48/13
- b) 48/13
- c) 52/13
- d) -48/33

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```

struct Koordinata{unsigned char x,y;};
void main(int argc, char* argv[])
{
    Koordinata nuqta1={45,40}, nuqta2={'0',50};
    cout<<          (pow(nuqta1.y-nuqta2.x,2)-pow(nuqta1.y-
nuqta2.x,2))<<endl;
}

```

- a) 0
- b) 1
- c) 4
- d) 9

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```

struct Ishchi{unsigned int Yoshi; float Vazni;};
int main(int argc, char* argv[])
{
    Ishchi Eshmat={20,77.8},Toshmat={25,100};
    if(Eshmat.Vazni<Toshmat.Vazni) cout<<"Eshmat!"<<endl;
    else cout<<"Toshmat!"<<endl;
    return 0;
}

```

- a) Eshmat!
- b) Toshmat!
- c) 20,77.8
- d) 25,100

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Racional{int surat,maxraj};  
void main(int argc, char* argv[])  
{  
Racional a={5,3}, b={2,4};  
a.surat=(a.surat+b.maxraj)/a.maxraj;  
b.surat=(b.surat+a.maxraj)/a.surat;  
cout<<b.maxraj<<"/"<<a.maxraj+a.surat<<endl;  
}
```

- a) 4/6
- b) 7/3
- c) 2/3
- d) 0.3

## Амалий машғулот 28-29.

### Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ўқиш ва ёзиш функциялари

**Ишдан мақсад:** C++ дастурлаш тилида файллар билан ишлаш, файл ва сатр оқимлари устида турли амаллар бажариш, форматли ўқиш ва ёзиш функцияларидан фойдаланиш кўникмаларига эга бўлиш. Дастурда форматли ўқиш ва ёзиш функцияларидан фойдалана олиш.

**Масаланинг қўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

#### Ишни бажариш учун намуна

**Мисол:**  $f_1$  ва  $f_2$  матн файлларида бутун сонлар камаймайдиган тартибда киритилган. Ушбу файллар элементларидан, сонларнинг тартибланиши сақланган ҳолда,  $f_3$  матн файли ҳосил қилинсин. Файлларнинг дискдаги номлари буйруқ сатри орқали киритилсин.

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;
const int n = 80;
typedef FILE * Fmatn;
typedef char Satr[n];
int Ulash(Satr f1_nomi, Satr f2_nomi, Satr f3_nomi)
{
    int a, b;
    bool tamom_f1 = false, tamom_f2 = false;
    Fmatn f1, f2, f3;
    if ((f1 = fopen(f1_nomi, "rt")) == NULL)
    {
        cout << "" << f1_nomi << "' fayli ochilmadi!";
    }
}
```



```

        return 1;
    }
    if ((f2 = fopen(f2_nomi, "rt")) == NULL)
    {
        cout << "'" << f2_nomi << "' fayli ochilmadi!";
        return 1;
    }
    if ((f3 = fopen(f3_nomi, "wt")) == NULL)
    {
        cout << "'" << f3_nomi << "' fayli hosil qilinadi!";
        return 1;
    }
    if (fscanf(f1, "%i", &a) == EOF) tamom_f1 = true;
    if (fscanf(f2, "%i", &b) == EOF) tamom_f2 = true;
    do
    {
        while (!tamom_f1 && (tamom_f2 || a <= b))
        {
            fprintf(f3, "%i ", a);
            if (fscanf(f1, "%i", &a) == EOF) tamom_f1 = true;
        }
        while (!tamom_f2 && (tamom_f1 || b < a))
        {
            fprintf(f3, "%i ", b);
            if (fscanf(f2, "%i", &b) == EOF) tamom_f2 = true;
        }
    } while (!tamom_f1 || !tamom_f2);
    fclose(f1);
    fclose(f2);
    fclose(f3);
    return 0;
}

```

```

int main(int argc, char* argv[]){
    Satr f1_nomi, f2_nomi, f3_nomi;
    if (argc != 4)
    {
        cout << "Buyruq satri noto'g'ri kiritilgan!";
        return 0;
    }
    strcpy(f1_nomi, argv[1]);
    strcpy(f2_nomi, argv[2]);
    strcpy(f3_nomi, argv[3]);
    if (!Ulash(f1_nomi, f2_nomi, f3_nomi))
        cout << "\nFayl hosil qilindi: f1+f2=>f3";
    else
        cout << "\nMasalani echish imkoniyati yo'q.";
    system("pause");
    return 0;
}

```

Дастур ишлаши натижаси:

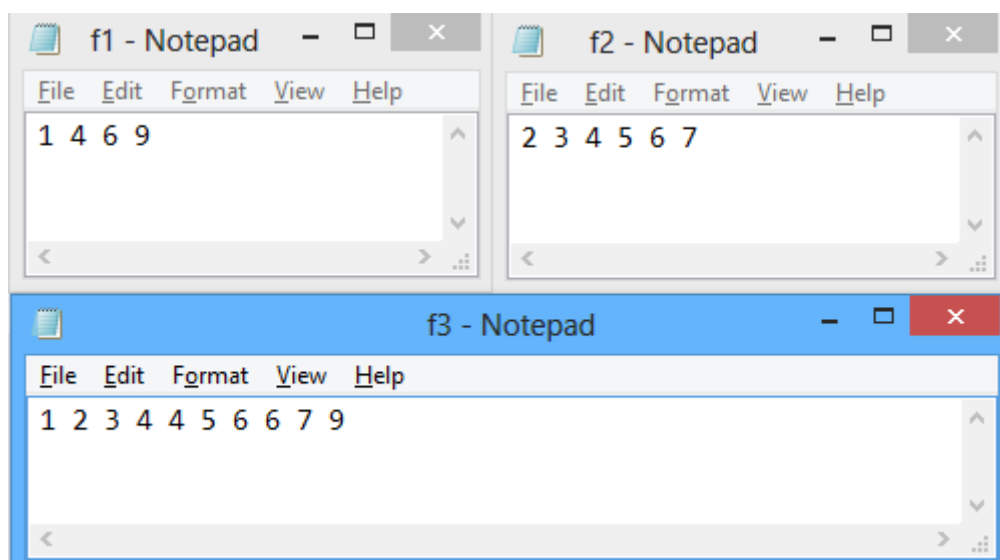
```

C:\windows\system32\cmd.exe - Project10.exe f1.txt f2.txt f3.txt
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Baron>cd Documents
C:\Users\Baron\Documents>cd "Visual Studio 2013"
C:\Users\Baron\Documents\Visual Studio 2013>cd Projects
C:\Users\Baron\Documents\Visual Studio 2013\Projects>cd Project10
C:\Users\Baron\Documents\Visual Studio 2013\Projects\Project10>cd Debug
C:\Users\Baron\Documents\Visual Studio 2013\Projects\Project10\Debug>Project10.e
xe f.txt ff.txtfff.txt
Buyruq satri noto'g'ri kiritilgan!
C:\Users\Baron\Documents\Visual Studio 2013\Projects\Project10\Debug>Project10.e
xe f1.txt f2.txt f3.txt

Fayl hosil qilindi: f1+f2=>f3
Press any key to continue . . .

```



### Амалий топшириқлар

1. Берилган  $s$  матн файлининг манфий элементлари йиғиндисини ҳисоблайдиган функция тузилсин.
2. Бўш бўлмаган, бутун сонлардан ташкил топган  $r$  матн файлдаги сонлар ўсиш ёки камайиш тартибида жойлашганлигини текширадиган мантиқий функция тузилсин.
3. Берилган  $t_1$  ва  $t_2$  матн файллар бири иккинчисининг нусхаси ёки йўқлигини текширадиган мантиқий функция тузилсин.
4. Камида иккита сатрдан иборат бўлган  $f$  матн файлининг охиридан битта олдинги сатрини натижа сифатида қайтарадиган функция тузилсин.
5. Берилган  $s$  сатрда учраган рақамларни  $t$  матнга ўтказадиган функция тузилсин.
6. Берилган мусбат бутун  $n$  сонидан ошмайдиган Фибоначчи сонларини  $f$  матн файлига ёзадиган функция тузилсин.
7. Бутун сонлар ёзилган, бўш бўлмаган  $f$  матн файл берилган. Файлдаги элементларнинг ўрта арифметигидан кичик бўлган файл элементлари миқдорини аниқлайдиган функция тузилсин.
8. Берилган  $f$  матн файли бўш бўлмаган  $n$  сатрдан иборат. Сатрдаги сўзлар бири-биридан вергул билан ажратилган, охириги сўз нукта билан тугаган. Ҳар бир сатрнинг охириги сўзидан фарқ қилувчи сўзлардан ташкил топган сатр  $g$  матн файлга ёзилсин.
9. Берилган  $t$  матн файл учун қуйидаги функциялар тузилсин:
  - а)  $\text{add1}(t,s)$ ,  $t$ -матн бошига  $s$  сатрни қўшадиган;
  - б)  $\text{addlast}(t,s)$ ,  $t$ -матн охирига  $s$  сатрини қўшадиган;
  - д)  $\text{double}(t)$ ,  $t$ -матндаги ҳар бир рақамни иккилантирадиган;
  - е)  $\text{replace}(t,s)$  - бўш бўлмаган  $t$  матннинг охириги сатрини  $s$  сатри билан алмаштирадаган;

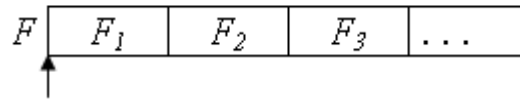
- ф) next(t) - t матнда учраган ҳар бир рақамни, шу рақамдан кейин келувчи рақам билан алмаштирадиган ('9' рақами '0' билан алмаштирилади);
- ж) del(t)- t матндаги охиридан битта олдинги сатрини (агар у мавжуд бўлса) ўчира-диган;
- и) first(t) - t матнда ҳар бир сатрнинг фақат биринчи учраганини қолдирадиган.
10. Ҳақиқий сонлар ёзилган f матн файлидаги энг узун ўсувчи кетма-кетлик элементлари миқдорини аниқловчи функция тузилсин.
11. Берилган f ва g матн файлларида сонлар камаймайдиган кўринишда тартибланган бўлсин. Бу файлларни ягона камаймайдиган кўринишда h файлга бирлаштириш талаб қилинади.
12. Мантиқий relation(f) функцияси тузилсин. Бу функция f файлдаги берилганлар “муносабат” синтаксисидаги тўғри ёзув эканлиги аниқласин (қоидалар қуйида келтирилган).
- <муносабат> ::= <сон> <муносабат белгиси> <сон>  
 <муносабат белгиси> ::= <= | > | <= | <> | >= |  
 <сон> ::= <рақам> | <рақамлар>  
 <рақамлар> ::= <нол эмас> <рақам> | <рақамлар> <рақам>  
 <рақам> ::= 0 | <нол эмас>  
 <нол эмас> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
13. Тўққизта сатрлардан иборат t матн файлини ҳосил қилувчи функция тузилсин. Бунда 1-сатрда битта '1' белгиси, 2-сатрда иккита '2' белгиси ва шу тариқа 9-сатрда тўққизта '9' белгиси бўлсин.

### Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

### 1. Файл кўрсаткичи



### 2. Матн файлнинг хотирада ва амалда кўриниши.

1- satr#13#10	2- satr uzunroq #13#10	#13#10	4-satr#13#10#26
---------------	------------------------	--------	-----------------

Тўртта сатрдан ташкил топган матн файли

Унинг кўриниши:

1- satr[13][10]

2- satr uzunroq[13][10]

[13][10]

4- satr[13][10]

[26]

### 3. Форматли ўқиш ва ёзишда ишлатиладиган тур аломатлари.

Тур аломати	Кутилаётган киймат	Аргумент тури
<b>Сон туридаги аргумент</b>		
d, D	Ўнлик бутун	int * arg yoki long * arg
E, e	Сузувчи нуқтали сон	float * arg
F	Сузувчи нуқтали сон	float * arg
G, g	Сузувчи нуқтали сон	float * arg
O	Саккизлик сон	int * arg
O	Саккизлик сон	long * arg
I	Ўнлик, саккизлик ва ўн олтилик бутун сон	int * arg
I	Ўнлик, саккизлик ва ўн олтилик бутун сон	long * arg
U	Ишорасиз ўнлик сон	Unsigned int * arg
U	Ишорасиз ўнлик сон	Unsigned long * arg
x	Ўн олтилик сон	int * arg
X	Ўн олтилик сон	int * arg

Белгилар		
S	Сатр	char * arg (белгилар массиви)
C	Белги	char * arg (белги учун майдон кенглиги берилиши мумкин (масалан, %4s). Н белгидан ташкил топган белгилар массивига кўрсаткич: char arg [N])
%	'%' белгиси	Ҳеч қандай алмаштиришлар бажарилмайди, '%' белгиси сақланади.
Кўрсаткичлар		
N	int * arg	%n аргументигача муваффақиятли ўқилган белгилар сони, айнан шу инт кўрсаткичи бўйича адресда сақланади.
P	YYYY:ZZZZ ёки ZZZZ кўринишидаги ўн олтилик	Объектга кўрсаткич (far* ёки near*).

### Кейслар банки

**Кейс 1.** Тузилмалар билан ишлайдиган дастур тузилди. Дастур вазифаси китоб тузилмаси орқали муаллифнинг китоблари сони бўйича муаллифлар фамилия исмларини ўсиш бўйича тартиблаб чиқаришдан иборат. Дастур ишга туширилганда тартиблаш жараёни бажарилмаётгани маълум бўлди. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш бошқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тилида дастур коддини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
string s="Salom", s1="talaba";  
int n=5;  
string s2=s+s1;  
cout<<s2.at(n);
```

- a) t
- b) a
- c) m
- d) b

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
string s1="AB", s2="BCC", s3="CAC";  
s2.erase(1,2);  
s1.insert(1,s3);  
s2.append(s3);  
s3.erase(2,1);  
s3=s1.substr(1,2)+s2.substr(1,1);  
cout<<s3;
```

- a) BAC
- b) SAC
- c) ABC
- d) ACB

3. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int a = 2, b = a + 3;  
do {  
    a = 12;  
    b++;  
    a = a + b;  
}  
while (a <= b);  
cout<<a;
```

- a) 18
- b) 21
- c) 15
- d) 19

4. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s[] = "O'zbekiston Milliy Universiteti talabalari";  
char c='a'; int k=0;  
for(int i=0; i<strlen(s); i++)  
    if(s[i]==c) k++;  
cout<<k;
```

- a) 4
- b) 3
- c) 5
- d) 2

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
string s1="AB", s2="BCC", s3="CAC";  
int i=1;  
s1.insert(i, s3);  
s2.append(s3);  
s2.erase(1, 2);  
s3.erase(2, 1);  
s3=s1+s2.substr(1, 1);  
cout<<s3;
```

- a) ACACBC
- b) ABVCC
- c) ACACBVC
- d) ACBACBC



6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="BAB", s2[]="ABC", s3[]="ABA";  
string a="lalalal";  
if (strrev(s3)==s3)  
cout<<strrev(strcpy(s2, strrev(s1)));  
else cout << a.find("al");
```

- a) BAB
- b) ABA
- c) ABC
- d) CBA

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int s=0, m=0 ;  
char matn[] = "Programmash asoslari fani 2015 - yil";  
for(int i=0;i<strlen(matn);i++)  
    if(isdigit(matn[i])) s++;  
    else if (isalpha(matn[i])) m++;  
cout<<s<<"\t"<<m;
```

- a) 4 28
- b) 2 31
- c) 5 29
- d) 6 23

8. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
int s=0, m=0 ;  
char matn[] = "Programmash asoslari - c++,c#  
programmash tillari: ";  
for(int i=0;i<strlen(matn);i++)  
    if(!isdigit(matn[i])&& !isalpha(matn[i]))  
        if(matn[i]!=' ')  
            cout<<matn[i]<<" ";
```

- a) - +, #:
- b) - ++, #:
- c) - + a + :, #:
- d) - + s, #:

9. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Uchburchak{int a,b,c};
void main(int argc, char* argv[])
{
Uchburchak b={1,2,2}, a={33,3,7},c=b;
int yuza=a.a*a.b*c.c;
int yza=a.a+b.b+a.c;
cout<<yuza;
}
```

- a) 198
- b) 221
- c) 189
- d) 212

10. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
string s="Satrlar", s1="Belgilar", s3="Dastur";
    s.swap(s1);
    s.erase(0,5);
    s1.assign(s);
    cout<<s3.append(s1);
```

- a) Dasturlar
- b) Satrlar
- c) Belgilar
- d) BelgiDastur

## Амалий машғулот 30-31.

### Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари

**Ишдан мақсад:** C++ дастурлаш тилида сатрлар билан ишлаш, сатр устида турли амаллар бажариш, сатр функцияларидан фойдаланиш кўникмаларига эга бўлиш. Дастурда турли узунликдаги сатрлардан (ASCIIZ сатрлар, string туридаги сатрлар) фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** C++ тилидаги турларни уларнинг ҳажми ва максимал кийматларини файлга ёзадиган дастур тузилсин.

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;
#include <cstring>
int main()
{
    ofstream fout("turlar.txt", ios_base::out |
ios_base::trunc);
    if (!fout.is_open())
    {
        cout << "Fayl ochilmadi yoki yaratilmadi\n";
        return 1;
    }
    fout << "    tur nomi    " << "byte" << "    " <<
"max qiymat " << endl;
    fout << "bool    = " << sizeof(bool) << "
" << fixed << setprecision(2)
```

```

        << (pow(2, sizeof(bool)* 8.0) - 1) << endl;
    fout << "char          = " << sizeof(char) << "
" << fixed << setprecision(2)
        << (pow(2, sizeof(char)* 8.0) - 1) << endl;
    fout << "short int      = " << sizeof(short int) <<
"
" << fixed << setprecision(2)
        << (pow(2, sizeof(short int)* 8.0 - 1) - 1) << endl;
    fout << "unsigned short int = " << sizeof(unsigned short
int) << "
" << fixed << setprecision(2)
        << (pow(2, sizeof(unsigned short int) * 8.0) - 1) <<
endl;
    fout << "int          = " << sizeof(int) << "
" << fixed << setprecision(2)
        << (pow(2, sizeof(int)* 8.0 - 1) - 1) << endl;
    fout << "unsigned int    = " << sizeof(unsigned int)
<< "
" << fixed << setprecision(2)
        << (pow(2, sizeof(unsigned int)* 8.0) - 1) << endl;
    fout << "long int       = " << sizeof(long int) << "
" << fixed << setprecision(2)
        << (pow(2, sizeof(long int)* 8.0 - 1) - 1) << endl;
    fout << "unsigned long int = " << sizeof(unsigned long
int) << "
" << fixed << setprecision(2)
        << (pow(2, sizeof(unsigned long int) * 8.0) - 1) <<
endl;
    fout << "float         = " << sizeof(float) << "
" << fixed << setprecision(2)
        << (pow(2, sizeof(float)* 8.0 - 1) - 1) << endl;
    fout << "long float     = " << sizeof(long float) <<
"
" << fixed << setprecision(2)
        << (pow(2, sizeof(long float)* 8.0 - 1) - 1) <<
endl;
    fout << "double        = " << sizeof(double) << "
" << fixed << setprecision(2)
        << (pow(2, sizeof(double)* 8.0 - 1) - 1) << endl;;

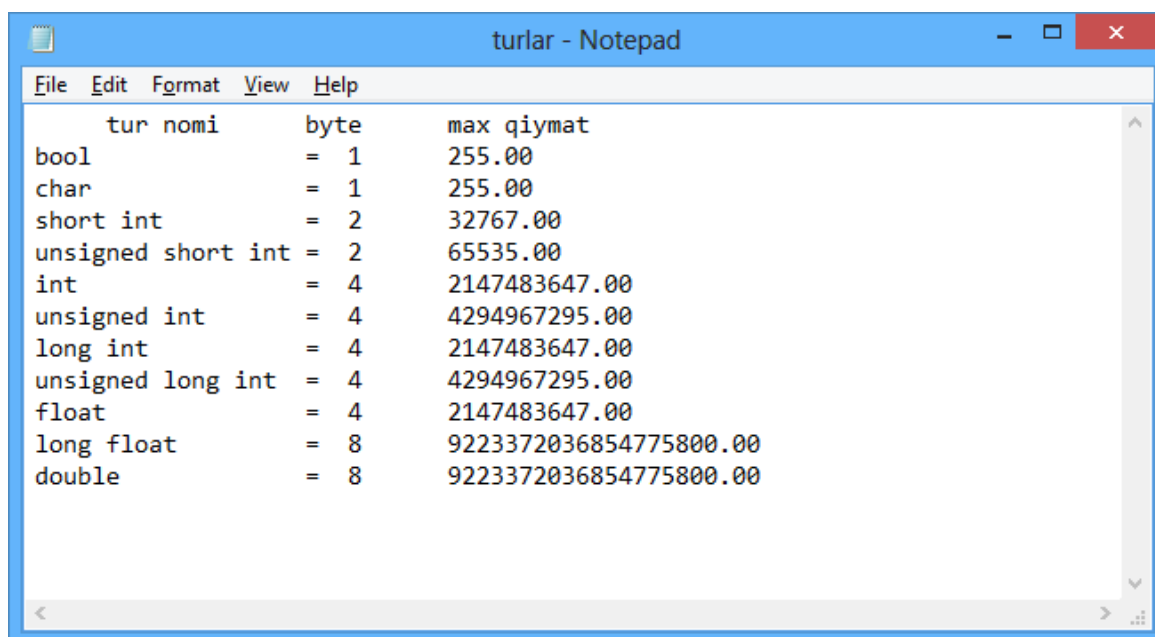
```

```

fout.close();
cout << "Ma'lumotlar turlar.txt fayliga muvaffaqiyatli
yozildi\n";
system("pause");
return 0;
}

```

Дастур ишлаши натижаси:



tur nomi	byte	max qiymat
bool	= 1	255.00
char	= 1	255.00
short int	= 2	32767.00
unsigned short int	= 2	65535.00
int	= 4	2147483647.00
unsigned int	= 4	4294967295.00
long int	= 4	2147483647.00
unsigned long int	= 4	4294967295.00
float	= 4	2147483647.00
long float	= 8	9223372036854775800.00
double	= 8	9223372036854775800.00

### Амалий топшириқлар

1. Матн файли ва к бутун сони берилган. Матн файлдаги к-сатр ўчирилсин. Агар файлда бу сатр бўлмаса, у ўзгармасдан қолдирилсин.
2. Матн файли ва к бутун сони берилган. Матн файлдаги к-сатр олдига (кейинга) бўш сатр қўйилсин. Агар файлда бу сатр бўлмаса у ўзгармасдан қолдирилсин.
3. Матн файли ва s сатри берилган. Файлдаги барча бўш сатрлар s сатр билан алмаштирилсин.
4. Матн файли берилган. Ундаги кетма-кет келувчи пробеллар битта пробел билан алмаштирилсин.
5. Ичидаги сатрлари 60 белгидан ошмаган ва чап томонга текисланган матн файли берилган. Ҳар бир бўш бўлмаган сатрлар олдига этарли сондаги пробелни қўйиш орқали сатрлар ўнг томонга (марказга) текислансин.
6. Берилган матн файлидаги абзатс бошланишини 5 та пробелдан бошланувчи сатр аниқлайди. Матн файли абзатс олдидаги пробеллар олиб ташлаш ва сатр олдига бўш сатрни қўйиш орқали янги кўринишга келтирилсин.

7. Берилган матн файлидаги сатрларни тескари тартибда жойлаштириш орқали янги матн файли ҳосил қилинсин.
8. Матн файли ва бутун к сони берилган. Матн файлидаги к- абзатс ўчирилсин. Абзатс бошида 5 та пробел билан бошланадиган сатр билан аниқланади. Агар бу тартиб номерли абзатс бўлмаса файл ўзгаришсиз қолдирилсин.
9. Ҳар бири чап ва ўнг томондан пробеллар билан тўлдирилган ҳақиқий сонни ифодаловчи сатрлардан ташкил топган матн файли берилган. Бу сонларнинг йиғиндиси ва миқдори аниқлансин. Масалани ечишда фақат файлдан белгили ўқиш амалга оширилсин.
10. Name1 ва Name2 номли матн файллари берилган. Улардаги мос тартиб номердаги сатрларни кетма-кет ёзишдан янги Name3 файли ҳосил қилинсин (Name1 файлининг 1-сатри, Name2 файлининг 1-сатри, Name1 файлининг 2-сатри, Name2 файлининг 2-сатри ва ҳоказо). Агар бирорта файл иккинчисидан қисқа бўлса, у тугаши билан иккинчи файлнинг қолган сатрлари Name3 файлига кўчирилади.
11. Иккита матн файли берилган. Уларнинг фақат биттасига кирувчи сатрлар чоп этилсин.
12. Пробел билан ажратилган, учтагача сон ёзилган сатрлардан иборат Name файли берилган. Янги Name1, Name2 ва Name3 матн файллари ҳосил қилинсинки, улар мос равишда ҳар бир сатрдаги учта сонларнинг ёзувларидан (сатростиларидан) ташкил топсин. Агар Name файли сатрида мос ўринда сон бўлмаса (“\*” ёзилган), улар ўрнига файлга “NULL” сатри ёзилсин.
13. Name1 номли бўш бўлмаган матн файли ва к натурал сони берилган. Иккита матн файллари ҳосил қилинсин: Name2 файлга Name1 ҳар бир сатрнинг биринчи (охирги) к та белгисини ўзида сақловчи, (агарда сатр узунлиги к кичик бўлса сатрнинг ўзини сақласин), Name3 файли эса ҳар бир сатрнинг к - белгисидан ташкил топсин (агарда сатр узунлиги к дан кичик бўлса Name3 файлига пробел ёзилсин).
14. Иккита  $f_1$  ва  $f_2$  матн файлларини сатрма-сатр солиштирадиган функция тузилсин. Функция  $f_2$  файлдаги  $f_1$  файлга нисбатан ўчирилган ёки кўшилган сатрларни чоп қилсин.
15. C++ тилидаги программадаги ҳар бир идентификатор ва у учраган сатрга кўрсаткич эслаб қолиш орқали идентификаторлар ва калит сўзлари луғати яратилсин.

## Адабиётлар рўйхати

1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

### 1. Файл очиш режимлари

Қиймати	Файл очилиш ҳолати тавсифи
r	Файл фақат ўқиш учун очилади
w	Файл ёзиш учун очилади. Агар бундай файл мавжуд бўлса, у қайтадан ёзилади (янгиланади).
a	Файлга ёзувни қўшиш режими. Агар файл мавжуд бўлса, файл унинг охирига ёзувни ёзиш учун очилади, акс ҳолда янги файл яратилади ва ёзиш режимида очилади.
r+	Мавжуд файл ўзгартириш (ўқиш ва ёзиш) учун очилади.
w+	Янги файл яратилиб, ўзгартириш (ўқиш ва ёзиш) учун очилади. Агар файл мавжуд бўлса, ундаги олдинги ёзувлар ўчирилади ва у қайта ёзишга тайёрланади.
a+	Файлга ёзувни қўшиш режими. Агар файл мавжуд бўлса, унинг охирига (EOF аломатидан кейин) ёзувни ёзиш (ўқиш) учун очилади, акс ҳолда янги файл яратилади ва ёзиш режимида очилади.

### 2. Оқимни очиш режимлари.

Очиш режими	Вазифаси
ios::app	Файл кўрсаткични файлни охирида жойлаштириб қўшиш режимида файлни очади.
ios::ate	Файл кўрсаткичини файлни охирига жойлаштиради. Ўқиш мумкин эмас, чиқарувчи маълумотлар файлни охирига ёзилади.

ios::in	Қиритиш учун файлни очилишини кўрсатади.
ios::nocreate	Агарда кўрсатилган файл мавжуд бўлмаса, файл яратилмайди ва хато қайтарилади.
ios::noreplace	Агарда файл мавжуд бўлса, очиш операцияси тўхташ ва хатоликни қайтариш лозим.
ios::out	Чиқариш учун файлни очилишини кўрсатади
ios::trunc	Мавжуд бўлган файлни ичидагини олиб ташлайди (кўчиради).
ios::binary	Файлни бинар файл кўринишида очади

### Кейслар банки

**Кейс 1.** Бинар файллар билан ишлайдиган дастур тузилди. Дастурнинг вазифаси файлдан китоб ҳақида маълумотларни олиш ва ёзишдан иборат. Файлдан маълумот ўқиб олиш жараёнида дастур хатолиги келиб чиқди. Дастурни тўғри ишлашини таъминлаш керак.

#### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тилида дастур коддини тақдим этинг (индивидуал ҳолда).

### Тест саволлари

1. C++ тилида матн файлини фақат ўқиш учун очиш ифодаси тўғри кўрсатилган жавобни топинг.

- FILE \*f=fopen("D:\\fayllar\\matn.txt", "rt")
- FILE f=fopen("D:\\fayllar\\matn.txt", "rt")
- FILE \*f=fopen("rt", "D:\\fayllar\\matn.txt")
- FILE \*f=fopen("D:\\fayllar\\matn.txt")

2. C++ тилида матн файлини фақат ёзиш учун очиш ифодаси тўғри кўрсатилган жавобни топинг.

- FILE \*f=fopen("D:\\fayllar\\matn.txt", "wt")
- FILE f=fopen("D:\\fayllar\\matn.txt", "wt")
- FILE \*f=fopen("wt", "D:\\fayllar\\matn.txt")
- FILE \*f=fopen("D:\\fayllar\\matn.txt")



3. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```
void f(int i, int*j, int &k)
{
i++;
(*j)++;
k++;
}
void main(){
fstream f("fayl.txt", ios::out);
int =1, j=2, k=3;
f(i,&j,k);
f << i << j << k;
}
```

- a) 1 3 4
- b) 2 3 4
- c) 1 2 3
- d) 2 2 4

4. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```
fstream f("fayl.txt", ios::out);
int a = 5, c = 7;
for(int i=0; i<c; i=i+2)
{
if(i%2==0) a++;
else if (i%3==0) a=a+2;
if (i%4==0) a=a+3;
i--;
}
f<<a;
```

- a) 17
- b) 25
- c) 19
- d) 21

5. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```
fstream f("fayl.txt", ios::out);
```

```
int a = -10, b = 4;
```

```
do
```

```
{
```

```
    a += 3;
```

```
    b++;
```

```
}
```

```
while (a == b);
```

```
f<<a;
```

a) -7

b) 12

c) -10

d) 4

6. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```
    ofstream f("fayl.txt", ios::out);
```

```
    const int n=3;
```

```
    int a[n][n]={{7,2,3},{3,2,4},{0,6,4}};
```

```
    int s=0;
```

```
    for(int i=0;i<n;i++)
```

```
        s+=a[0][i]+a[i][n-1];
```

```
    f<<s;
```

a) 23

b) 13

c) 18

d) 26

7. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```
struct Ishchi{unsigned int Yoshi; float Vazni;};  
void main()  
{  
fstream f("fayl.txt", ios::out);  
Ishchi Eshmat={20,77.8},Toshmat={23,80};  
Eshmat.Yoshi=Eshmat.Yoshi+Eshmat.Vazni/Toshmat.Vazni;  
f<<Eshmat.Yoshi+Toshmat.Vazni<<" yoshda!"<<endl;  
}
```

- a) 100 yoshda!
- b) 43 yoshda!
- c) 157 yoshda!
- d) 77 yoshda!

8. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```
fstream f("fayl.txt", ios::out);  
string s1="A", s2="B", s3="C";  
int i=1;  
s1.insert(i,s3);  
s1.append(s2);  
s1.insert(i,s3);  
s1.append(s2);  
f<<s1;
```

- a) ACCBVB
- b) ABVBCC
- c) BACCBV
- d) BCAAAC

9. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```
fstream f("fayl.txt", ios::out);
```

```
int a = 3, b = 10, c=b/a;
```

```
do
```

```
{
```

```
  a += b/c+1;
```

```
  b++;
```

```
  c=b/c;
```

```
}
```

```
while (a % 2 != 0);
```

```
f<<a;
```

a) 20

b) 30

c) 10

d) 25

10. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```
fstream f("fayl.txt", ios::out);
```

```
int a = 5, b = 8;
```

```
if (++a == b++)
```

```
  a += a;
```

```
if(a % 2 == 0)
```

```
  a += b * b;
```

```
else
```

```
  a += b;
```

```
f<<a;
```

a) 87

b) 13

c) 69

d) 33

## Амалий машғулот 32-33.

### Динамик тузилмалар

**Ишдан мақсад:** C++ дастурлаш тилида динамик тузилмалар билан ишлаш, чизикли рўйхат, стек, навбат устида турли амаллар бажариш, улардан фойдаланиш кўникмаларига эга бўлиш. Дастурда чизикли рўйхатлардан фойдалана олиш.

**Масаланинг кўйилиши:** Тингловчи вариант бўйича берилган масалани C++ дастурлаш тилида ишлаши ва керакли натижа олиши лозим.

### Ишни бажариш учун намуна

**Мисол:** Нолдан фарқли бутун сонлардан иборат чизикли рўйхат яратилсин ва ундан кўрсатилган сонга тенг элемент ўчирилсин.

Дастур коди:

dastur.cpp файли:

```
#include<iostream>
using namespace std;

struct Zanjir
{
    int element;
    Zanjir * keyingi;
};

Zanjir * Element_Joylash(Zanjir * z, int yangi_elem)
{
    Zanjir * yangi = new Zanjir;
    yangi->element = yangi_elem;
    yangi->keyingi = 0;
    if (z)
    {
        Zanjir * temp = z;
        while (temp->keyingi)
            temp = temp->keyingi;
```

```

        temp->keyingi = yangi;
    }
    else z = yangi;
    return z;
}
Zanjir * Element_Uchirish(Zanjir * z, int del_elem)
{
    if (z)
    {
        Zanjir * temp = z;
        Zanjir * oldingi = 0;
        while (temp)
        {
            if (temp->element == del_elem)
            {
                if (oldingi)
                {
                    oldingi->keyingi = temp->keyingi;
                    delete temp;
                    temp = oldingi->keyingi;
                }
                else
                {
                    z = z->keyingi;
                    delete temp;
                    temp = z;
                }
            }
            else
            {
                oldingi = temp;
                temp = temp->keyingi;
            }
        }
    }
}

```

```

    }
    return z;
}
void Zanjir_Ekranga(Zanjir * z)
{
    cout << "Zanjir elementlari: " << endl;
    Zanjir * temp = z;
    while (temp)
    {
        cout << temp->element << ' ';
        temp = temp->keyingi;
    }
    cout << endl;
}
Zanjir * Zanjirni_Uchirish(Zanjir * z)
{
    Zanjir * temp = z;
    while (z)
    {
        z = z->keyingi;
        delete temp;
    }
    return z;
}

```

```

int main(){
    Zanjir * zanjir = 0;
    int son, del_element;
    do
    {
        cout << "\nSonni kiriting(0 - jarayonni tugatish):";
        cin >> son;
    }
}

```

```

        if (son)zanjir = Element_Joylash(zanjir, son);
    } while (son);
Zanjir_Ekranga(zanjir);
cout << "\n O'chiriladigan elementni kiriting : ";
cin >> del_element;
zanjir = Element_Uchirish(zanjir, del_element);
Zanjir_Ekranga(zanjir);
zanjir = Zanjirni_Uchirish(zanjir);system("pause");
return 0;
}

```

Дастур ишлаши натижаси:

```

C:\Users\Baron\documents\visual studio 2013\Projects\Project10\Debug\Project...
Sonni kiriting<0 - jarayonni tugatish>: 4
Sonni kiriting<0 - jarayonni tugatish>: 6
Sonni kiriting<0 - jarayonni tugatish>: 3
Sonni kiriting<0 - jarayonni tugatish>: 8
Sonni kiriting<0 - jarayonni tugatish>: 4
Sonni kiriting<0 - jarayonni tugatish>: 2
Sonni kiriting<0 - jarayonni tugatish>: 0
Zanjir elementlari:
4 6 3 8 4 2

O'chiriladigan elementni kiriting : 4
Zanjir elementlari:
6 3 8 2

```

### Амалий топшириқлар

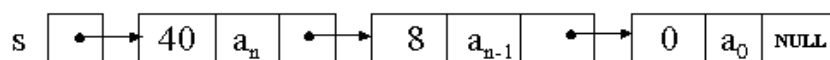
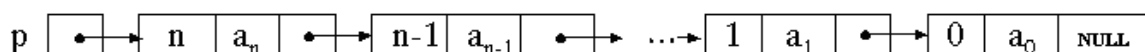
1.  $Z$  занжир  $\text{int}$  туридаги элементга (элемент майдони) эга.  $Z$  занжир элементларининг ўрта арифметиғи топилсин.
2.  $Z$  занжири элементи  $\text{сатр}$  турида.  $Z$  занжирнинг охириги элементи билан устма-уст тушадиган элементлар сони аниқлансин.
3.  $Z$  занжир элементлари бутун турда.  $Z$  занжирдан иккита  $Z_1$  ва  $Z_2$  занжирлар ҳосил қилинсин. Бунда  $Z_1$ - занжирнинг мусбат элементлари ва  $Z_2$  - қолган элементлари.
4. Орага қўйишни бажарувчи функциялар тузилсин:
  - а) бўш бўлмаган  $Z$  занжирининг биринчи элементидан кейин янги у элементни;
  - б)  $Z$  занжирига кирувчи ҳар бир у элементдан кейин янги  $x$  элементни жойлаштирадиган.



5. Бўш бўлмаган ва тартибланган  $Z$  занжирга янги у элементни шундай жойлаштирингки, унда  $Z$  занжирдаги тартиб бузилмасин.
6. Занжирнинг элементларини ўчирадиган функция тузилсин:
  - а) бўш бўлмаган  $Z$  занжирнинг биринчи элементини;
  - б)  $Z$  занжирнинг иккинчи элементини, агар у мавжуд бўлса.
7. Бутун турдаги элементлардан иборат  $Z$  занжир элементини ўчирадиган функция тузилсин:
  - а) бўш бўлмаган  $Z$  занжирнинг охириги элементини;
  - б)  $Z$  занжирдан биринчи манфий элемент, агар у мавжуд бўлса;
  - д)  $Z$  занжирдаги барча манфий элементларни.
8. Клавиатурадан матнни (сатрни) ўқиб, тескари тартибда чоп қилинсин. Бунда матндаги ҳар бир сўз занжир элементи сифатида қаралсин.
9. Бутун  $n$  ( $n > 1$ ) ва  $n$  та ҳақиқий сонлар берилган. Бу сонлар камаймайдиган тартибда чоп қилинсин. Бунда ҳар бир сон занжир элементи сифатида қаралсин.
10. Қуйидаги амалларни бажарувчи функция тузилсин:
  - а)  $Z_1$  ва  $Z_2$  занжирлар тенглигини текширувчи;
  - б)  $Z_1$  занжир  $Z_2$  занжирга киришини аниқловчи.
11. Бутун коэффитсиентли

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + a_1 x + a_0$$

кўпхадни занжир кўринишида тасвирлаш мумкин (қуйидаги расмдаги  $n$  занжир), агар  $a_i = 0$  бўлса, мос халқа занжирга киритилмайди. Пастдаги расмда  $s$  занжир  $S(x) = 52x^{40} - 3x^8 + x$  кўпхадга мос занжир келтирилган.



Кўпхадни тавсифлашнинг занжир кўринишига мос турлар эълон қилинсин ва бу занжир устида қуйидаги амалларни бажарувчи функциялар тузилсин:

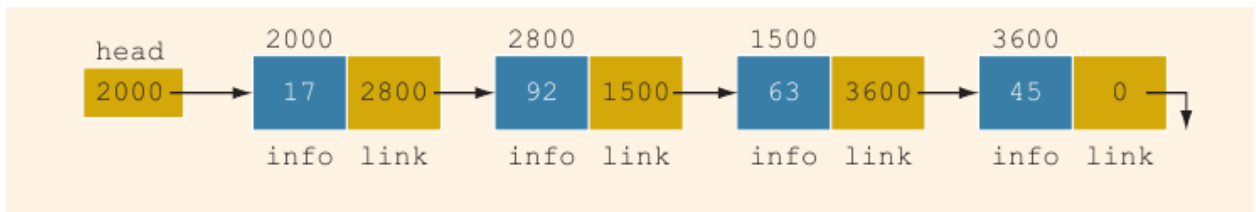
- а)  $p$  ва  $q$  кўпхадларни тенглигини текширувчи  $Teng(p,q)$  мантиқий функцияси;
- б)  $p$  кўпхаднинг  $x$ -бутун нуқтадаги қийматини ҳисобловчи  $Qiymat(p,x)$  функцияси;
- д)  $p$  кўпхаднинг ҳосиласи бўлган  $q$  кўпхадни курувчи  $Hosila(p,q)$  функцияси;

## Адабиётлар рўйхати

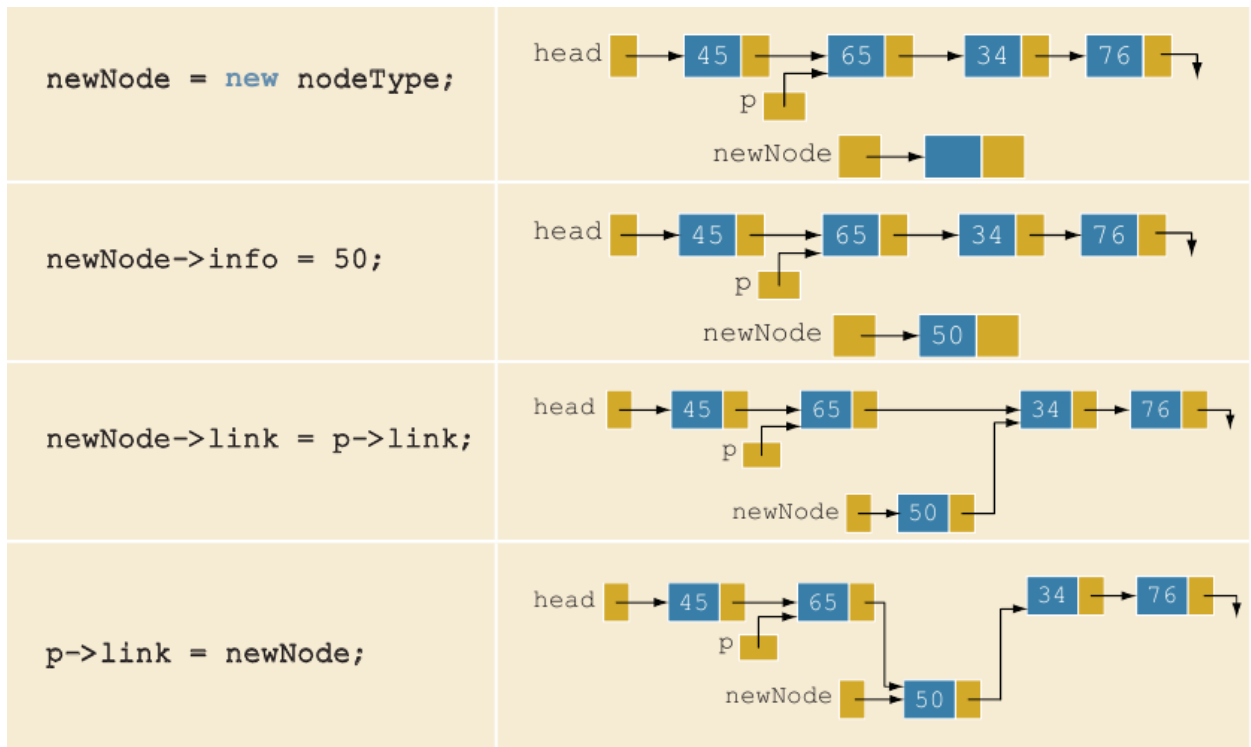
1. [D.S. Malik. C++ Programming: From problem analysis to program design. Course Technology, 2011.](#)
2. [Ivor Horton. Beginning Visual C++ 2005. Wiley Publishing, 2005.](#)
3. [Б. Страуструп. Язык программирования C++. Специальное издание.-М.: ООО «Бином-Пресс», 2006.](#)
4. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида Дастурлаш асослари// Тошкент, Ўзбекистон Миллий Университети, 2009.

## Тарқатма материаллар.

### 1. Чизикли рўйхат кўриниши



### 2. Чизикли рўйхатга элемент жойлаш.



## Кейслар банки

**Кейс 1.** Стеklar билан ишлайдиган дастур тузилди. Дастурнинг вазифаси стекдаги энг катта қийматли элементни топиш ва уни ўчиришдан иборат. Дастур ишлаши жараёнида стек бўшаб қолди. Дастурни тўғри ишлашини таъминлаш керак.

### Кейсни бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- C++ дастурлаш тилида турли усулларда дастур кодини тақдим этинг (индивидуал ҳолда).

## Тест саволлари

1. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="ABBA", s2[]="BC", s3[]="CCBA";  
strncat(s2,s3,2);  
strcpy(s1,s2);  
cout << s2;
```

- a) ABC
- b) BCCC
- c) CBVV
- d) BCCA

2. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
char s1[]="BAB", s2[]="ABC", s3[]="CBA";  
string a="ABC";  
if (s1[0]==s1[1] && s2[0])  
cout<<s3;  
else cout << a[1];
```

- a) B
- b) C
- c) A
- d) AB

3. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```
void f(int i, int*j, int &k)
{
i++;
(*j)++;
k++;
}
void main(){
fstream f("fayl.txt", ios::out);
int i=1, j=2, k=3;
f(i, &j, k);
f << i << j << k;
}
```

- a) 1 3 4
- b) 2 3 4
- c) 1 2 3
- d) 2 2 4

4. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```
fstream f("fayl.txt", ios::out);
int a = 5, c = 7;
for(int i=0; i<c; i=i+2)
{
if(i%2==0) a++;
else if (i%3==0) a=a+2;
if (i%4==0) a=a+3;
i--;
}
f<<a;
```

- a) 17
- b) 25
- c) 19
- d) 21

5. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Uchburchak{int a,b,c;};  
void main(int argc, char* argv[])  
{  
Uchburchak b={1,2,2}, a={33,3,7}, c=b;  
int yuza=a.a*a.b*c.c;  
int yza=a.a+b.b+a.c;  
cout<<yuza;  
}
```

- a) 198
- b) 212
- c) 205
- d) 172

6. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Vaqt{unsigned int soat, minut, sekund;};  
int main(int argc, char* argv[])  
{  
Vaqt a={2,34,11}, b={1,12,5}; b.minut=a.minut+9;  
a.sekund=b.sekund+a.minut;  
cout<<b.minut<<":"<<a.sekund<<":"<<b.soat<<endl;  
return 0;  
}
```

- a) 43:39:1
- b) 50:39:1
- c) 43:42:1
- d) 43:39:14

7. Қуйидаги программа қисми ишлаши натижасида экранга қандай қиймат чиқади?

```
struct Racional{int surat,maxraj;};  
void main(int argc, char* argv[])  
{
```

```

Racional a={52,33}, b=a;
a.maxraj=13;
b.surat=(a.surat+4)/a.maxraj-a.surat;
a.surat=(a.surat+b.maxraj)/b.surat;
cout<<b.surat<<"/"<<a.maxraj<<endl;
}

```

- a) -48/13
- b) 48/13
- c) 52/13
- d) -48/33

8. Қуйидаги дастур кодининг натижасини топинг.

```

num2 = 4 / 5 - 4 % 5;
cout<< num2 <<endl;

```

- a) -4
- b) 4
- c) 5
- d) 11

9. Қуйидаги дастур кодининг натижасини топинг.

```

num2 = 4 *3 + 7 / 5 - 25.5;
cout<< num2 <<endl;

```

- a) -12.5
- b) 10.5
- c) -12
- d) 12

10. Қуйидаги программа қисми ишлаши натижасида файлга қандай қиймат ёзилади?

```

fstream f("fayl.txt", ios::out);
int a = 5, b = 8;
if (++a == b++)
    a += a;
if(a % 2 == 0)
    a += b * b;

```

```
else
    a += b;
f<<a;
a) 87
b) 13
c) 69
d) 33
```