

**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA
MAXSUS TA‘LIM VAZIRLIGI**
O‘RTA MAXSUS, KASB-HUNAR TA‘LIMI MARKAZI

M.J. YUNUSOVA, A.B. RAHIMOV

DASTURLASH BO‘YICHA PRAKTIKUM

Kasb-hunar kollejlari uchun o‘quv qo‘llanma

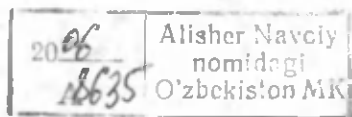


Toshkent — «ILM ZIYO» — 2006

*O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligi
O'rta maxsus, kasb-hunar ta'limi Markazining ilmiy-metodik
kengashi tomonidan nashrga tavsiya etilgan.*

O'quv qo'llanma hozirgi kunda kasb-hunar kollejlarda «Dasturlash» fanida rejalashtirilgan Turbo Paskal dasturlash tilida dasturlar tuzishga bag'ishlangan masalalar to'plami — praktikumdan iborat. Uning birinchi qismida Turbo Paskalda dastur tuzish uchun kerakli bilimlar, o'rnatilgan funksiya va protseduralar ro'yxati, o'zgaruvchilarning xotirada saqlanish sohasi, EHMda Turbo Paskal tilida yozilgan dasturlar bilan ishlash tartibi to'g'risidagi eng zaruriy nazariy yordam ma'lumotnomalari berilgan. Ikkinchi qismda masalalar shartlari keltirilgan. Ular quyi va yuqori murakkablikdagi masalalarni o'z ichiga olgan ikki bo'limga bo'lingan. Masalalar oddiydan murakkabga tomon, kasb-hunar kollejlarda Turbo Paskalda dasturlash fani o'quv rejasi mavzulariga mos ravishda o'zgarib boradi. Uchinchi qismda ikkinchi qismda berilgan masalalarning dasturlari keltirilgan.

Taqrizchilar: **A. A. XOLYIGITOV** — fizika-matematika fanlari doktori, professor; **J. JUMAYEV** — Buxoro Davlat universitetining dotsenti; **M.M. XUDOYQULOV** — Buxoro kommunal xo'jaligi kasb-hunar kollejining direktori.



10 32166
39,



KIRISH

Zamonaviy EHMlar (kompyuterlar) o'zining avvalgi avlodlari kabi ishlash chog'ida eng oddiy arifmetik va mantiqiy amallarni bajara oladi, xolos. Mashina murakkab masalani yechishi uchun uning xotirasiga mos dastur kiritilgan bo'lishi kerak. Dastur buyruqlar ketma-ketligidan, ko'rsatmalar — biror oddiy amalni bajarishga oid mashinaga beriladigan ko'rsatmalardan iborat bo'lib, ancha murakkab amalni ifodalaydi.

Mazkur o'quv qo'llanma kasb-hunar kolleji talabalarida, shuningdek, dasturlash bilan qiziquvchilarda dasturlash bo'yicha boshlang'ich ko'nikmalar hosil qilishga bag'ishlangan. EHMlarga dastur-topshiriq yozish uchun istalgan dasturlash tilini tanlab olish mumkin. Kitobda shunday til sifatida deyarli barcha dasturlash tillarida uchraydigan tushuncha va tuzilishlarni o'z ichiga olgan Turbo Paskal tili tanlab olingan. Bu esa kitobxonga keyinchalik boshqa tillar bilan tanishish chog'ida dasturlash borasidagi bilimlarni kengaytirishga imkon beradi.

Dasturlashni o'rganish uchun dasturlash tiliga bag'ishlangan kitobni o'qib chiqish yetarli emas, buning uchun masalalarning dasturlarini yozish kerak. Bu kitobda boshlovchi dasturchi uchun u yecha olishi mumkin bo'lgan masalalar to'plami keltirilgan.

O'quv qo'llanma uch qismdan iborat. I qismda Turbo Paskal dasturlash tili bo'yicha, dasturlashda asqotadigan eng zaruriy tushunchalarni o'z ichiga olgan yordam ma'lumotnomasi keltirilgan. II qismda masalalar shartlari berilgan. Ular quyi va yuqori murakkablikdagi masalalarga ajratilgan. Quyi murakkablikdagi masalalar mavzulariga ko'ra guruhlangan. Yuqori murakkablikdagi masalalar esa mavzusiga emas, murakkabligiga ko'ra birlashtirilgan. Ular orasida har xil mavzularga tegishlilari bo'lib, ular, asosan, iqtidorli talabalarga mo'ljallab berilgan. III qismda masalalar asosiy qismining dastur matnlari keltirilgan, ularni o'rganish, dasturchiga, agar u masalani mustaqil yechgan bo'lsa ham, foyda keltirishi, shubhasiz.

Yuqori murakkablikdagi masalalarning faqat yechish dasturlarigina emas, balki ularni yechish algoritmlari ham bayon etilgan. Ular dasturlarni o'qishga yordam beradigan darajada ishlab chiqilgan. Masalani yechishga o'tishdan oldin talaba tegishli mavzuni o'rganishi uchun darslikni o'qib chiqishi kerak. Agar masalaning yechimi darrov topilmasa, uning yechimini ko'rish va yana bir marta masalani mustaqil yechishga urinish mumkin. Masala shartida bayon etilganidek, dastur ishlasa, demak, masala yechilgan hisoblanadi.



1.1. TURBO PASCALDA DASTUR TUZISH UCHUN KERAKLI MA'LUMOTLAR



1.1.1. Dastur tasnifi

Pascal tilidagi dastur quyidagi bo'limlardan iborat:

- nishona (metka)larni e'lon qilish bo'limi;
- o'zgarma(lar)larni e'lon qilish bo'limi;
- o'zgaruvchilarni e'lon qilish bo'limi;
- protsedura va funksiyalarni e'lon qilish bo'limi;
- dastur ko'rsatmalari bo'limi.

Umumiy holda dastur tasnifi quyidagi ko'rinishga ega:

label {metkalarni e'lon qilish}
const {o'zgarma(lar)larni e'lon qilish}
type {turlarni e'lon qilish}
var {o'zgaruvchilarni e'lon qilish}
{dasturchining protsedura va funksiyalarini e'lon qilish}
begin
{asosiy dastur ko'rsatmalari}
end.



1.1.2. Ma'lumotlarning asosiy turlari

Pascal tilining asosiy turlariga quyidagilar kiradi:

- butun sonlar (INTEGER va boshqalar);
- haqiqiy sonlar (REAL va boshqalar);
- belgilar (CHAR);
- satrlar (STRING);
- mantiqiy (BOOLEAN).

Butun va suzuvchi nuqtali sonlar har xil formatlarda berilishi mumkin.

Butun sonlar

Format	Qiyamatlar sohasi
SHORTING	— 128 — 127
INTEGER	— 32768 — 32767

LONGINT	- 2147483648 - 2147483647
BYTE	0 - 255
WORD	0 - 65535

Haqiqiy sonlar

Format	Qiymatlar sohasi	Qiymati
REAL	2.9E - 39 - 1.7e 38	11-12
SINGLE	1.5E - 45 - 3.4e 38	7 - 8
DOUBLE	5.0E - 324 - 1.7e 108	15 - 16
EXTENDED	3.4E - 4932 - 1.1e 4932	- 19 - 20

Satrlar

Satrlar ikki xil ko'rinishda e'lon qilinishi mumkin:

1-ko'rinish:

Ism: *String*.

255 belgidan iborat uzunlikka ega o'zgaruvchi e'lon qilinadi.

2-ko'rinish:

Ism: *String* [satr uzunligi].

Ko'rsatilgan uzunlikdagi o'zgaruvchi e'lon qilinadi.



1.1.3. Massivlar

Bir o'lchovli massivni e'lon qilish:

Massiv ismi: *array* [quyi indeks .. yuqori indeks] *of* <elementlar turi>.

Ikki o'lchovli massivni e'lon qilish:

Massiv ismi: [quyi indeks 1 .. yuqori indeks 1, quyi indeks 2 .. yuqori indeks 2] *of* <elementlar turi>.



1.1.4. Yozuvlar

Yozuvlarni ikki xil ko'rinishda e'lon qilish mumkin:

1-ko'rinish:

O'zgaruvchi Yozuv: *record*

Maydon 1 : 1 tur;

Maydon 2 : 2 tur;

....

Maydon 3 : 3 tur;

end.

2-ko'rinish:

Avval yozuv turi, keyin o'zgaruvchi tur e'lon qilinadi:
type

Tur Yozuvi Ismi: *record*

Maydon 1 : tur 1;

Maydon 2 : tur 2;

Maydon *J* : tur *J*;

end;

var.

O'zgaruvchi Yozuv : Tur Yozuvi Ismi.



1.1.5. KO'RSATMALAR (operatorlar)

TANLASH ko'rsatmalari

1) IF ko'rsatmasi

1-ko'rinish:

if shart

then

begin

{bu ko'rsatmalar, agar shart to'g'ri bo'lsa, bajariladi}

end

else

begin

{bu ko'rsatmalar shart xato bo'lsa, bajariladi}

end.

2-ko'rinish:

IF shart

then

begin

{bu ko'rsatmalar shart to'g'ri bo'lganda bajariladi}

end.



Izoh : agar *begin* va *end* o'rtasida faqat bitta ko'rsatma (operator) bo'lsa, *begin* va *end*ni yozmaslik mumkin.

2) CASE ko'rsatmasi

1-ko'rinish:

Case ifoda *of*

begin

```

1 — o'zgarmlar ro'yxati:   begin
                               {1-ko'rsatmalar}
                               end;
2 — o'zgarmlar ro'yxati:   begin
                               {2-ko'rsatmalar}
                               end;
...
J — o'zgarmlar ro'yxati:   begin
                               {J-ko'rsatmalar}
                               end;
else
                               begin
                               {ko'rsatmalar}
                               end;
end.

```

Case so'zidan keyin keluvchi ifoda qiymati, tegishli ro'yxat o'zgarmlari bilan mos kelgandagina *begin* va *end* o'rtasidagi ko'rsatmalar bajariladi, aks holda *else* so'zidan keyin keluvchi *begin* va *end* so'zlari o'rtasidagi ko'rsatmalar bajariladi.

SIKLLAR

3) FOR ko'rsatmasi

1-ko'rinish (hisobchi qiymati o'sadi):

```

for Hisobchi: = Boshlang'ich Qiymat to So'nggi Qiymat do
    begin
        {ko'rsatmalar}
    end;

```

begin va *end* o'rtasidagi ko'rsatmalar [(So'nggi Qiymat — Boshlang'ich Qiymat) + 1] marta bajariladi.

Agar boshlang'ich qiymat so'nggi qiymatdan katta bo'lsa, *begin* va *end* o'rtasidagi ko'rsatmalar bajarilmaydi.

! *Izo h.* agar *begin* va *end* o'rtasida faqat bitta ko'rsatma bo'lsa, *begin* va *end* so'zlarini yozmaslik mumkin.

2-ko'rinish (hisobchi qiymati kamayadi):

```

for Hisobchi: = Boshlang'ich Qiymat downto So'nggi Qiymat do
    begin
        {ko'rsatmalar}
    end;

```

begin va *end* o'rtasidagi ko'rsatmalar [(Boshlang'ich Qiymat — So'nggi Qiymat) + 1] marta bajariladi. Agar boshlang'ich qiymat so'nggi qiymatdan kichik bo'lsa, *begin* va *end* o'rtasidagi ko'rsatmalar bajarilmaydi.

4) REPEAT ko'rsatmasi

repeat

begin

{ko'rsatmalar}

end

until Shart;

begin va *end* o'rtasidagi ko'rsatmalar (sikl ko'rsatmalari) bajariladi, shundan keyin shart ifodaning qiymati tekshiriladi. Agar u *False* ga teng bo'lsa (ya'ni, shart bajarilmasa), sikl ko'rsatmasi yana bir marta bajariladi. Xuddi shunday tarzda shart haqiqat bo'lguncha davom ettiriladi. Shunday qilib, *until* so'zidan keyin siklni tugatish sharti yoziladi.



Izoh: agar *begin* va *end* o'rtasida faqat bitta ko'rsatma yozilgan bo'lsa, *begin* va *end* so'zlarini yozmaslik mumkin.

5) WHILE ko'rsatmasi

while Shart *do*

begin

{shartlar}

end.

Shart ifodasidagi qiymat tekshiriladi, agar u *Truega* teng bo'lsa (shart bajarilsa), *begin* va *end* o'rtasidagi ko'rsatmalar (sikl ko'rsatmalari) bajariladi. Shundan keyin yana Shart ifodasining qiymati tekshiriladi, shunday tarzda Shart ifodasining qiymati *False* bo'lguncha davom ettiriladi. Shunday qilib, *while* so'zidan keyin sikl ko'rsatmalarining bajarish sharti yoziladi.



Izoh: agar *begin* va *end* o'rtasida faqat bitta ko'rsatma yozilgan bo'lsa, *begin* va *end* so'zlarini yozmaslik mumkin.

6) GOTO shartsiz o'tish ko'rsatmasi

GoTo metka (nishona)

Oldida, *label* bo'limida e'lon qilingan, metka yozilgan ko'rsatmaga o'tish amalga oshiriladi.

1.1.6. Funksiyalarni e'lon qilish

Function Funksiya Ismi (*var* Parametr 1: Tur 1;
var Parametr 2: Tur 2;

...
var Parametr J : Tur J) : Tur;

const
{o'zgarmaslarni bayon etish}
var
{o'zgaruvchilarni bayon etish}
begin
{funksiyalar ko'rsatmalari}
Funksiya Ismi: = qiymati;
end.

1.1.7. Protseduralarni e'lon qilish

Procedure Protsedura Ismi (*var* Parametr 1 : Tur 1;
var Parametr 2 : Tur 2 . . .
var Parametr J : Tur J);

const
{o'zgarmaslarni bayon etish}
var
{o'zgaruvchilarni bayon etish}
begin
{protsedura ko'rsatmalari}
end.

1.2. TURBO PASCAL REZERV SO'ZLARI, DIREKTIVALARI, STANDART FUNKSIYA VA PROTSEDURALARI

1.2.1. Rezerv so'zlar va direktivalar

Turbo Pascal tilining rezerv so'zlari

<i>and</i>	<i>file</i>	<i>not</i>	<i>them</i>
<i>array</i>	<i>for</i>	<i>object</i>	<i>to</i>
<i>asm</i>	<i>function</i>	<i>of</i>	<i>type</i>
<i>begin</i>	<i>go to</i>	<i>or</i>	<i>unit</i>
<i>case</i>	<i>if</i>	<i>packed</i>	<i>until</i>
<i>const</i>	<i>implementation</i>	<i>procedure</i>	<i>uses</i>
<i>constructor</i>	<i>in</i>	<i>program</i>	<i>var</i>
<i>destructor</i>	<i>inherited</i>	<i>record</i>	<i>while</i>

<i>div</i>	<i>inline</i>	<i>repeat</i>	<i>with</i>
<i>do</i>	<i>interface</i>	<i>set</i>	<i>xor</i>
<i>downto</i>	<i>label</i>	<i>shl</i>	
<i>else</i>	<i>mod</i>	<i>shr</i>	
<i>end</i>	<i>nil</i>	<i>string</i>	



1.2.2. Turbo Paskal direktivalari

<i>absolute</i>	<i>external</i>	<i>forward</i>	<i>near</i>	<i>public</i>
<i>assembler</i>	<i>far</i>	<i>interrupt</i>	<i>private</i>	<i>virtual</i>



1.2.3. Funksiya va protseduralar

Matematik	Izoh
ABS (ifoda)	Argumentning mutlaq qiymati (butun yoki haqiqiy tur)
SQR (ifoda)	Argument kvadrati (butun yoki haqiqiy tur)
SQRT (ifoda : <i>real</i>) : <i>real</i>	Argument kvadrat ildiz chiqarish
SIN (ifoda : <i>real</i>) : <i>real</i>	Radianlarda ifodalangan argumentning sinusi
COS (ifoda : <i>real</i>) : <i>real</i>	Radianlarda ifodalangan argumentning kosinusi
ARCTAN (ifoda : <i>real</i>) : <i>real</i>	Radianlarda ifodalangan argumentning arktangensi
EXP (ifoda : <i>real</i>) : <i>real</i>	Argument eksponentasi
LN (ifoda : <i>real</i>) : <i>real</i>	Argumentning natural logarifmi
INT (ifoda : <i>real</i>) : <i>real</i>	Argument butun qismi
TRUNC (ifoda : <i>real</i>) : <i>longint</i>	Argument butun qismi
ROUND (ifoda : <i>real</i>) : <i>longint</i>	Argument qiymatini eng yaqin butun songacha yaxlitlash
STR (ifoda : <i>var</i> Satr: <i>string</i>)	Sonli ifodani satrga aylantirish
Val (Satr : <i>string</i> ; var O'zgaruvchi, var Xato kodi : <i>integer</i>)	Butun yoki haqiqiy sonni ifodalovchi satrni songa aylantirish

Satr va belgilar bilan ishlash uchun	Izohlar
Concat (Satr 1 : <i>string</i> ,...; Satr N : <i>string</i>) : <i>string</i>	Bir necha satrni bittaga birlashtirish
Copy (Satr: <i>string</i> , Belgi Raqami : <i>integer</i>) : <i>string</i>	Ichki satrni ajratish
Delete (var Satr: <i>string</i> , Belgi Raqami : <i>integer</i> , Qancha : <i>integer</i>)	Satr qismini olib tashlash
Length (Satr : <i>string</i>) : <i>integer</i>	Satr uzunligi
Pos (Satr : <i>string</i> , Ichki satr : <i>string</i>) : <i>byte</i>	Ichki satrning satrdagi o'rni
Chr (Belgi Kodi : <i>byte</i>) : <i>Char</i>	Ko'rsatilgan koddagi belgi

Funksiya va protseduralar bayonida quyidagicha belgilashlar qabul qilingan:

- funksiya va protseduralar ismi yarim yo'g'on shrift bilan ajratilgan;
- parametrlar kursiv bilan ajratilgan; (parametr sifatida o'z-garmaslar, o'zgaruvchilar yoki tegishli turdagi ifodalar ishlatilishi mumkin. Agar parametr sifatida asosiy dasturning o'zgaruvchisi bo'lishi shart bo'lsa, uning oldiga, albatta, *var* so'zi, parametrdan keyin uning turi yozilishi kerak);
- shart bo'lmagan parametrlar kvadrat qavslarda yozilgan;
- funksiya parametrlari ro'yxatidan keyin (ikki nuqta orqali) funksiya qaytaradigan natija turi ko'rsatiladi.



1.2.3.1. Matematik funksiyalar

1. *Abs*

Sintaksis:

function Abs (x).

Butun yoki haqiqiy turdagi argumentning mutlaq qiymatini qaytaradi.

2. *Arctan*

Sintaksis:

function Arctan (x : real) : real.

Radianlarda ifodalangan argument burchak miqdorining arktangensini hisoblaydi.

3. *Cos*

Sintaksis:

function Cos (x : real) : real.

Radianlarlarda ifodalangan argument burchak miqdorining kosinusini hisoblaydi.

4. *EXP*

Sintaksis:

function EXP (x : real) : real.

Argument eksponentasiga teng qiymatni hisoblaydi.

5. *Ln*

Sintaksis:

function Ln (x : real) : real.

Argumentning natural logarifmiga teng bo'lgan qiymatni hisoblaydi.

6. *Sin*

Sintaksis:

function Sin (x : real) : real.

Radianlarda ifodalangan argument burchak miqdorining sinusini hisoblaydi.

7. *SQR*

Sintaksis:

function SQR (x).

Butun yoki haqiqiy turdagi ifoda argumentining kvadratini hisoblaydi.

8. *SQRT*

Sintaksis:

function SQRT (x : real) : real.

Argumentdan kvadrat ildiz chiqarilgan qiymatni aniqlaydi.

9. *Random*

Sintaksis:

function Random [(coha : word)].

Agar Soha parametri ko'rsatilmagan bo'lsa, $0 \leq x \leq 1$ shartni qanoatlantiruvchi X tasodifiy sonni beradi. Agar Soha parametri ko'rsatilgan bo'lsa, funksiya $0 \leq x \leq$ Soha shartni qanoatlantiruvchi *Word* turidagi tasodifiy sonni qabul qiladi.



Izoh: *Random* funksiyasiga birinchi marta murojaat qilishda, *Randomize* protsedurasini chaqirish yordamida tasodifiy sonlar dastur generatorini initsializatsiya qilish zarur.

10. *Randomize*

Sintaksis:

procedure Randomize.

Tasodifiy sonlar dastur generatorini initsializatsiyalaydi.



1.2.3.2. O'zgartirishning funksiya va protseduralari

1. *INT*

Sintaksis:

function Int (x : real) : real.

Haqiqiy turdagi qiymat sifatida argumentga butun qismni beradi. Shakl almashtirishda argumentning kasr qismi hisobga olinmaydi, ya'ni funksiya yaxlitlamaydi.

2. *Round*

Sintaksis:

function Round (x : real) : real.

Argumentni eng yaqin butun songa yaxlitlaydi.

3. *Str*

Sintaksis:

procedure Str (x [:2 Belgilar Jami [:Kasr Qism]]; *var Satr* : string).

Sonli ifodani uning satrli bayoniga aylantiradi. *Belgilar Jami* va *Kasr Qism* — butun turdagi belgilarning umumiy sonini va son tasviridagi kasr qism belgilari miqdorini beruvchi, shart bo'lmagan, ifodadir.

4. *Trunc*

Sintaksis:

function Trunc (x : real) : longint.

Argumentning butun qismini butun turdagi qiymat sifatida beradi. Shakl almashtirishda argumentning kasr qismi hisobga olinmaydi, ya'ni funksiya yaxlitlanmaydi.

5. *Val*

Sintaksis:

Procedure Val (*Satr* : string; *var O'zgaruvchi*; *var Xato* : integer).

Butun yoki haqiqiy sonni ifodalovchi satrni songa aylantiradi. hosil bo'lgan qiymat protsedurani chaqirishda ko'rsatilgan o'zgaruvchi tomonidan o'zlashtiriladi.

«Xato» o'zgaruvchi, agar shakl almashtirish bajarilmasa, uning amalga oshmasligiga sabab bo'lgan satr belgisi raqamini o'zlashtiradi. Agar shakl almashtirish muvaffaqiyat bilan bajarilsa, «Xato»ning qiymati nolga teng.



1.2.3.3. *Satr va belgilar bilan ishlash uchun funksiyalar va protseduralar*

1. *Chr*

Sintaksis:

function Chr (Belgi Kodi : byte) : char.

Ko'rsatilgan kod bilan belgini qabul qiladi.

2. *Concat*

Sintaksis:

function Concat ($S1$ [, $S2$, ..., SN] : string) : string.

Funksiyani chaqirishda ko'rsatilgan qatorlar birlashmasi bo'lgan satrni qabul qiladi.

3. *Copy*

Sintaksis:

function Copy (s : string; n : integer; l : integer) : string.

S satr qismi ichki satrni qabul qiladi. Ichki satr n —raqamli belgidan boshlanadi va l belgidan iborat bo'ladi.

4. *Delete*

Sintaksis:

procedure Delete (var *S*: string; Belgi Raqami: integer; Qancha: integer).

S satrdan *n* raqami bilan boshlanuvchi va *l* belgidan iborat bo'lgan qismni o'chiradi.

5. *Length*

Sintaksis:

function Length (satr : string) : integer.

Argument — satr belgilari miqdoriga teng qiymatni qaytaradi.

6. *POS*

Sintaksis:

function POS (satr : string; Ichki satr : string) : byte.

Satrdagi ichki satr o'rnini (belgi raqamini) qaytaradi.

7. Satr va belgilar ishlatilgan dasturlarda ko'pincha grafik rejimning *DetectGraph* protsedurasi ishlatiladi.

DetectGrap

Sintaksis:

DetectGrap (var *Drayver*, *Rejim* : integer).

Grafik adaptorni tekshiradi va qaysi grafik drayveri va rejim ishlatilayotganini aniqlaydi.



1.2.3.4. Fayllar bilan ishlash uchun protsedura va funksiyalar

1. *Append*

Sintaksis:

procedure Append (var *F* : text).

Fayl oxiriga qo'shish rejimida faylli o'zgaruvchi *F* bilan bog'liq mavjud faylni ochadi. Mavjud bo'lmagan faylni ochishga urinishda dasturni bajarish vaqti xatosi paydo bo'ladi.

2. *Assing*

Sintaksis:

procedure Assing (var *F* ; Fayl Ismi : string).

F fayl o'zgaruvchini aniq fayl bilan bog'laydi.

3. *Close*

Sintaksis:

procedure Close (var *F*).

F fayl o'zgaruvchi bilan bog'liq bo'lgan faylni yopadi.

4. *EOF*

Sintaksis:

function EOF (var *F*) : boolean.

F fayl o'zgaruvchi bilan bog'liq fayldan o'qishda faylning oxiriga (*end of file*) yetganligini tekshiradi. Agar o'quv ko'rsatkichi fayl oxiriga yetgan bo'lsa, funksiya *EOF* — *TRUE*, aks holda — *FALSE* qiymatni oladi.

5. *Erase*

Sintaksis:

procedure Erase (var F).

Ismi *F* fayl o'zgaruvchi bilan bog'liq faylni yo'qotadi.

6. *IOResult*

Sintaksis:

function IOResult : integer.

Oxirgi bajarilgan faylni kiritish-chiqarish (shu jumladan faylni ochish va yopish) ishi natijasi kodini qabul qiladi. Agar kiritish-chiqarish ishi muvaffaqiyatli bajarilgan bo'lsa, funksiya nolni qaytaradi.



Izoh: dastur *IOResult* funksiyasini ishlata olishi uchun kiritish-chiqarish natijasida xato paydo bo'lishi mumkin bo'lgan ko'rsatmalar oldida {1 □ -}, ko'rsatmalardan keyin esa {□ 1 +} direktivalarni joylashtirish kerak.

7. *Reset*

Sintaksis:

procedure Reset (var F [: file; Recsize : word]).

Mavjud faylni ochadi. Fayl turi ixtiyoriy bo'lishi mumkin. Agar fayl elementlari standart turlarning birortasiga tegishli bo'lmasa, *Recsize* parametr yozuv uzunligini (baytlarda) beradi. Mavjud bo'lmagan faylni ochishga urinishda bajarish vaqti xatosi paydo bo'ladi.

8. *Rewrite*

Sintaksis:

procedure Rewrite (var F [: file; Recsize : word]).

Ismi *F* fayl o'zgaruvchi bilan bog'liq bo'lgan faylni yaratadi va ochadi. Agar shunday ismli fayl bor bo'lsa, *Rewrite* protsedura uni yo'qotadi va yangisini yaratadi.



1.2.3.5. Boshqa funksiyalar

1. *ClrEol*

Sintaksis:

procedure ClrEol.

Joriy (kursor turgan) satrni, kursor turgan xonadan satr oxirigacha *TextBackGrand* protsedurasi bergan rang bilan bo'yab, tozalaydi.

2. *ClrScr*

Sintaksis:

procedure ClrScr.

Ekranning joriy oynasini (*Windows* protsedurasi o'rnatgan) *TextBackGrand* protsedurasi bergan rang bilan bo'yab, tozalaydi.

3. *Delay*

Sintaksis:

procedure Delay (To'xtatish : word).

Dastur bajarilishini millisekundlarga to'xtatishni ta'minlaydi.

4. *Dispose*

Sintaksis:

procedure Dispose (var P).

Dinamik o'zgaruvchi egallagan *P* ko'rsatayotgan xotirani bo'shatadi.

5. *Eoln*

Sintaksis:

function Eoln [(var F : text)] : boolean.

Agar *Eoln* funksiyaga murojaat qilinganda, parametr ko'rsatilmagan bo'lsa, funksiya klaviaturaning buferidan o'qilayotgan navbatdagi belgi «yangi satr» belgisi bo'lish-bo'lmasligini tekshiradi. Agar bo'lsa, funksiya *TRUE*, aks holda *FALSE* qiymatlarini qabul qiladi. Agar parametr ko'rsatilgan bo'lsa, funksiya xuddi shunday tartibda, funksiyani chaqirishda ko'rsatilgan, fayl o'zgaruvchi bilan bog'liq bo'lgan matnli faylning navbatdagi belgisini tekshiradi.

6. *GotoXY*

Sintaksis:

procedure GotoXY (x, y : byte).

(*x, y*) koordinataga ega bo'lgan ekranning nuqtasiga kursorni ko'chiradi.

7. *Halt*

Sintaksis:

procedure Halt [(Tugatish kodi : word)].

Dastur bajarilishini tugatadi va boshqarishni operatsion tizimga uzatadi.

8. *New*

Sintaksis:

procedure New (var P).

Dinamik o'zgaruvchi uchun xotira ajratadi va *P* ko'rsatkich uchun ajratilgan soha adresini beradi.

9. *ParamCount*

Sintaksis:

function ParamCount : word.

Buyruq satriga parametrlar miqdorini beradi.

10. *ParamStr*

Sintaksis:

function ParamStr (N : word) : string.

Funksiyaga murojaat qilganda raqami ko'rsatilgan buyruq satri parametrini qabul qiladi. Bajarilayotgan dastur fayliga yo'l va uning ismi (masalan, s: *tp\exe@tpu\myprog.exe*) *ParamStr* (0)ning qiymati bo'ladi.

11. *ReadKey*

Sintaksis:

function ReadKey : char.

Bosilgan klavishga mos belgini beradi. *Ord* funksiyasidan foydalanib, bosilgan klavish kodini hosil qilish mumkin. Agar xizmat klavishi bosilgan bo'lsa (masalan, kursorni siljitish klavishi), *ReadKey* funksiya 0 kodlik belgini beradi. Bu holda xizmat klavishini o'xshatish uchun yana bir marta *ReadKey* funksiyani chaqirish kerak, bu holda u xizmat kodi klavishini beradi. Quyida tez-tez ishlatiladigan ba'zi bir klavishlarning kodlari keltirilgan:

Klavish	Kod
<Esc>	27
<Backspace>	8
<Enter>	13
<↑>	0; 72
<↓>	0; 80
<←>	0; 75
<→>	0; 77

12. *TextBackGround*

Sintaksis:

procedure TextBackGround (rang : byte).

Write va *writeln* ko'rsatmalari bilan chiqariladigan axborotlar tag rangini beradi. Rang parametri sifatida quyida nomlangan o'zgarma(lar)ning birontasi ishlatilishi mumkin:

O'zgarma	Rang	Rang raqami
<i>Black</i>	Qora	0
<i>Blue</i>	Ko'k	1
<i>Green</i>	Yashil	2
<i>Cyan</i>	Firuza	3
<i>Red</i>	Qizil	4
<i>Magenta</i>	Binafsha	5
<i>Brown</i>	Jigarrang	6
<i>Light Gray</i>	Och kulrang	7

13. TextColor

Sintaksis:

procedure TextColor (rang : byte).

Write va *writeln* ko'rsatmalari bilan chiqariladigan axborot belgilarining rangini o'rnatadi. Rang parametri sifatida quyida keltirilgan nomlangan o'zgarma(lar)ning biridan foydalanish mumkin:

O'zgarma	Rang	Rang raqami
<i>Black</i>	Qora	0
<i>Blue</i>	Ko'k	1
<i>Green</i>	Yashil	2
<i>Cyan</i>	Firuza	3
<i>Red</i>	Qizil	4
<i>Magenta</i>	Binafsha	5
<i>Brown</i>	Jigarrang	6
<i>Light Gray</i>	Och kulrang	7
<i>Dark Gray</i>	To'q kulrang	8
<i>Light Blue</i>	Havorang	9
<i>Light Green</i>	Och yashil	10
<i>Light Cyan</i>	Och firuza	11
<i>Light Red</i>	Och qizil (alvonrang)	12
<i>Light Magenta</i>	Och binafsha	13
<i>Yellow</i>	Sariq	14
<i>White</i>	Oq yarqiroq	15

14. Where X

Sintaksis:

function Where X : byte.

Window protsedurasi bilan berilgan kursorning joriy oynadagi *X* koordinatasini ochadi.

15. Where Y

Sintaksis:

function Where Y : byte.

Window protsedurasi bilan berilgan kursorning joriy oynadagi *X* koordinatasini ochadi.

16. Window

Sintaksis:

procedure Window (x1, y1, x2, y2 : byte).

Ekran sohasi — oynani aniqlaydi. *x1, y1* koordinatalar oynaning yuqori chap burchagi koordinatalarini, *x2, y2* esa oynaning o'ng past burchagi koordinatalarini beradi.

1.3. O'ZGARUVCHILARNING XOTIRADA SAQLANISHI

Kompyuter xotirasida sonlar bitlar ko'rinishida saqlanadi. Har bir bit bitta ikkilik raqamiga teng qiymat (0 yoki 1) qabul qiladi. Sakkiz bit bir baytga teng. Sakkizta ikkilik raqam yordamida yoziladigan maksimal son — 11111111, bu 255 soniga mos, eng kichik son esa nol. Shuning uchun bayt 0 dan 255 gacha qiymat qabul qilishi mumkin.

O'zgaruvchilar xotirada saqlanadi. Har xil turdagi o'zgaruvchilar har xil qiymatlar qabul qilishi mumkinligi uchun ularni saqlashga xotiraning har xil sig'imi kerak. O'zgaruvchi uchun xotira baytlarga ajratiladi (butun son). Masalan, char turidagi o'zgaruvchi qiymat 256 ta belgidan ixtiyoriy bittasi bo'lishi mumkin. Shuning uchun bunday turdagi o'zgaruvchini saqlash uchun bitta bayt yetarlik. *Integer* butun turdagi o'zgaruvchining qiymati bo'lib — 32768 dan 32767 (65535 ta qiymat) gacha son bo'lishi mumkin. Bu turdagi o'zgaruvchini saqlash uchun 2 bayt kerak. Demak, tur qiymati sohasi qancha katta bo'lsa, shu turdagi o'zgaruvchini saqlash uchun shuncha ko'p bayt kerak:

O'zgaruvchi tur	Egallaydigan xotira, bayt	Qiymatlar sohasi
CHAR	1	Ixtiyoriy belgi
STRING	256	256 tagacha belgiga ega satr
STRING [N]	$1 \times n$	n tagacha belgiga ega satr
BYTE	1	0—255
WORD	2	0—65535
INTEGER	2	—32768—32767
LONGINT	4	—2147483648—2147483647
REAL	6	$2.9E - 39 - 1.7e 38$
SINGLE	4	$1.5E - 45 - 3.4e 38$
DOUBLE	8	$5.0E - 324 - 1.7e 308$
EXTENDED	8	$3.4E - 4932 - 1.1e 4932$

Dasturda bir xil qiymat uchun har xil turdagi o'zgaruvchilardan foydalanish mumkin, bunda ular xotiraning har xil hajmini egal-laydi. Masalan, agar dasturda *Day* (oy kuni) o'zgaruvchisi ishla-tilayotgan bo'lsa, uning uchun *byte*, *integer* yoki *longint* turlarini berish mumkin. Birinchi holda xotiraning bir, ikkinchisida — ikki, uchinchisida — to'rt bayti band bo'ladi. Lekin faqat bitta baytdan to'liq foydalaniladi, qolganlari esa faqat band qilinadi. Shuning uchun har bir o'zgaruvchi uchun eng to'g'ri keladigan turni tanlash kerak. Satr o'zgaruvchilari va massivlarni bayon etishga alohida

e'tibor berish kerak. Satr o'zgaruvchilariga joy ajratilayotganda, agar satrning chegaraviy uzunligi berilmagan bo'lsa, o'zgaruvchiga 256 hayt ajratiladi. Masalan, o'zgaruvchini e'lon qilishda *name* : *string* deb emas, balki *name* : *string* [30] deb yozish kerak.

Dasturning har bir massivi uchun joy ajratiladi, joy hajmi massiv elementlari turi va ularning miqdori bilan aniqlanadi. Masalan, ikki o'lchamli (masalan, 20x20) massivning haqiqiy sonlarini saqlash uchun xotiraning 3 kilobaytdan ko'proq sig'imi kerak ($20 \times 20 \times 8 = 3200$).

Kompyuter xotirasi hajmi chegarasiz bo'lib ko'ringani bilan, u samarasiz ishlatilsa, yetishmay qolishi mumkin.

1.4. EHMDA PASKAL TILIDA YOZILGAN DASTURLAR BILAN ISHLASH

Bosh menyuga kirish:

F10 klavishi bosiladi.

Menyudan yoki muloqot oynasidan hech bir ish bajarmasdan chiqish:

ESC klavishi bosiladi.

Bosh menyuning ixtiyoriy ichki menyusini tez ochish: *Alt* klavishi bosiladi va ushlab turiladi, u qo'yib yuborilmasdan menyu nomlanishining birinchi harfi bosiladi. Masalan, *Alt + F* — ekranda *File* menyusini ochadi.

Menyu buyruqlarini tez chiqarish. Ochilgan menyu buyruqlarini zudlik bilan chiqarish uchun buyruq nomidagi rangi bilan ajralib turgan harf yozilgan klavishni bosish kerak. Masalan, ochilgan *File* menyusida «A» klavishi bosilsa, *Save AS...* buyrug'i chiqariladi. Bu ish tezkor klavishlar yordamida ham bajarilishi mumkin:

Alt + F A

Ishchi (joriy) katalogni o'zgartirish.

1. *File* menyusini ochiladi.
2. *Change dir...* buyrug'i chiqariladi. Muloqot oyna ochiladi, uning *Directory* name maydonida joriy katalog nomi, *Directory tree* maydonida esa joriy diskning kataloglar shajarasi ko'rsatilgan bo'ladi.
3. *Directory name* maydoniga yangi joriy katalog nomi yoziladi, *TAB* klavishi yozilgan kataloglar shajarasida uning ismi ko'rsatiladi.
4. Tanlangan katalog *Enter* klavishini bosish bilan mustahkamlanadi.

5. *TAB* klavishini bosish bilan kursor *OK* tugmaga o'rnatiladi va *<Enter>* bosiladi.

Yangi dasturni kiritish uchun oyna ochish:

1. *File* menyusi ochiladi.
2. *New* buyrug'i chaqiriladi.

Diskka yangi dasturni yozish:

1. *File* menyusi ochiladi.
2. *Save as...* buyrug'i chiqariladi. *Save FileAs* muloqot oynasi ochiladi, bu yerda *Save FileAs* maydoniga yangi dastur uchun fayl nomi biriktiriladi. Shuningdek, bu oynada yangi dasturni mavjud fayllardan birining nomi bilan atash uchun ham imkoniyat bo'ladi. Buning uchun *TAB* klavishini bosish (kursor *File* maydoniga o'tadi) va mavjud ro'yxatdan fayl ismini tanlash kerak.

3. *Enter* klavishi bosiladi. Mavjud ism bilan yozilayotganda tegishli ogohlantirish berilgan va faylni qayta yozish yoki uni tanlangan ism bilan yozishdan voz kechish imkoniyati bo'ladi.

Faylni tahrir qilish uchun ochish:

1. *File* menyusi ochiladi.
2. *Open...* buyrug'i chiqariladi (*F3* klavish ham bu ishni bajaradi).
3. Ekranida *Open a File* oynasi paydo bo'ladi. *Name* maydoni *Files* maydoniga chiqariladigan fayllarni tanlash uchun shablonga ega bo'ladi. Sukut holatida **.pas* shablони tavsiya etiladi. Tegishli shablondagi fayllar ro'yxati *Files* maydoniga, diskning joriy katalogida mavjud bo'lgan fayllardan tanlab chiqariladi.

4. Bu bosqichda quyidagi ishlar bajariladi:

- 4a. *TAB* klavishi bosiladi, kursor *File* maydoniga o'tadi, mavjud ro'yxatdan fayl nomi tanlanadi.

- 4b. *Name* maydonidagi shablon o'zgartiriladi, shunday keyin 4a bosqichdagi ish bajariladi.

- 4d. Ochiladigan faylning nomi bevosita *Name* maydoniga kiritiladi.

5. *Enter* klavishi bosiladi. 4a va 4b hollarda klavish ikki marta bosiladi.

Faol oynadagi dasturni eski nom bilan saqlash:

I usul. *F2* klavishi bosiladi.

II usul. 1. *File* menyusi ochiladi.

2. *Save* buyrug'i chaqiriladi.

Faol oynadagi dasturni bajarishga yuborish:

Ctrl + F9 klavishlari bosiladi.

Natijani chiqarish uchun ekranni tozalash:

Ekranni tozalash uchun dasturda quyidagilarni bajarish zarur:

- *uses* taklifida *Crt* standart moduli kiritiladi (*Uses Crt*);
- dasturning operator bloki boshida *ClrScr* protsedurasi chaqiriladi.

Misol:

```
Progam PR;  
Uses Crt;  
var A: integer;  
begin  
  ClrScr;  
  Readln (A);  
  Writeln ( A = ', A);  
end.
```

Yangi dasturni yaratish va xatolarini tuzatishda bajariladigan ishlarning namunaviy rejai

1. Yangi dastur kiritish uchun tahrirlashning yangi oynasi ochiladi.
2. Yangi dastur matni teriladi.
3. Dastur matni diskka yoziladi (yangi dastur matni ishga tushirilguniga qadar diskka yozilishi zarur).
4. Dastur bajarishga yuboriladi (*Ctrl + F9*).
5. Agar dasturda sintaksis xatolarga yo'l qo'yilgan bo'lsa, ekranda tegishli axborot paydo bo'lib, kursor esa xato joyni ko'rsatadi. Bunday paytda tahrir buyruqlari yordamida tuzatilishi va yana dastur bajarishga yuborilishi kerak.
6. Dastur natijalarini ko'rish (*Alt + F5*).
7. Agar xato natijalar olingan bo'lsa, yo'l qo'yilgan algoritmik xatolarni tuzatish va dasturni yana bajarishga yuborish kerak.
8. 4—7-bosqichlar to'g'ri yechimlar olinguncha takrorlanadi.
9. Tuzatilgan dastur diskda saqlanadi (*F2*).

Paskal tilidagi dastur bilan ishlashda tezkor klavishlar.

Umumiy qo'llanishdagi klavishlar:

F10 — bosh menyuga kirish.

ESC — muloqot yoki menyu oynasini yopish.

Alt + X — Paskal dasturi bilan ishlash oynasidan chiqish.

Ctrl + Break — ishga tushirilgan dasturni to'xtatib, boshlang'ich oynaga qaytish (ishga tushgan dastur qotib qolganda yoki yopiq siklga tushib qolinganda bu klavishlar bosiladi).

Print Screen — ekran nusxasini printerda chop etish.

Pause — ixtiyoriy klavish bosilguncha o'zgarayotgan tasvirni ekranda to'xtatib turish.

Yordam tizimi bilan ishlash klavishlari:

F1 — yordam tizimining shu daqiqada faol bo'lgan oyna yoki kursor ko'rsatayotgan menyu buyrug'i to'g'risidagi axborotni yoritadi.

F1 — (ikki marta) — yordam tizimidan foydalanish to'g'risidagi ko'rsatmani ekranga chiqaradi.

Ctrl + F1 — faol oynadagi kursor turgan ibora haqidagi axborotni ekranga chiqaradi.

Alt + F1 — *Help* oynasining oldingi oynasiga qaytish. Bu buyruqni ko'p marta bajarilishi teskari tartibda *Help*ning 20 tagacha oxirgi oynalarini chiqaradi.

Shift + F1 — yordam tizimida mavjud bo'lgan iboralarning alfavit tartibidagi ro'yxatiga ega bo'lgan *Index* oynasini ekranga chiqaradi.

Fayllarni ochish, saqlash va tahrir oynalari bilan ishlash klavishlari:

F2 — tahrirlashning faol oynasidagi dasturni diskdagi fayлга eski nom bilan saqlash.

F3 — tahrirlash va ishga tushirish uchun zarur bo'lgan faylni diskdan tanlash uchun *Open a File* muloqot oynasini chiqarish.

Alt + F3 — tahrirlashning faol oynasini yopish.

F6 — bir necha marta *F6* klavishning ketma-ket bosilishi ochilgan oynalar faolligi almashinuvining takrorlanishini keltirib chiqaradi.

Shift + F6 — *F6* klavishiga o'xshab ishlaydi, lekin oynalarni teskari ketma-ketlikda almashtirib boradi.

Alt + O — Paskal ishga tushirilgan lahzadan ochilgan hamma oynalar ko'rsatilgan *Window List* muloqot oynasini ochadi.

F5 — faol oynani to'la ekran o'lchamida ochadi, agar oyna shu o'lchamda bo'lsa, uning boshlang'ich holatini tiklaydi.

Ctrl + F5 — faol oyna o'lchamini va (yoki) uning ekrandagi o'mini o'zgartirish. Oynaning ekran bo'yicha siljishi yo'nalish klavishlari yordamida, o'lchamni o'zgartirish esa *Shift* va yo'nalish klavishlarining bir vaqtda bosilishi bilan bajariladi.

Dastur matni bo'lagi bilan ishlash klavishlari:

Shift + «yo'nalish klavishlari» — dastur bo'lagini ajratadi.

Shift + *Del* — dasturning ajratilgan bo'lagini o'chiradi va *Clipboard* buferiga joylashtiradi.

Ctrl + *Ins* — dasturning ajratilgan bo'lagini *Clipboard* buferiga nusxalaydi.

Shift + *Ins* — *Clipboard*dagi ajratilgan bo'lakni oynadagi kursor joylashgan xonaga qo'yadi.

Ctrl + *Del* — ajratilgan bo'lakni *Clipboard* buferiga uzatmasdan o'chiradi.

Alt + *BackSpace* — tahrirlashdagi oxirgi ishni bekor qiladi.

Kompilyatsiya va bajarishga yuborish klavishlari:

Alt + *F9* — tahrirlashning faol oynasidagi faylni kompilyatsiya (tarjima) qilish.

F9 — .*exe* faylini yaratish bilan ko'p modulli dasturni shartli kompilyatsiya qilish. Agar oxirgi kompilyatsiya vaqtidan ba'zi bir modullarga o'zgartirishlar kiritilgan bo'lsa, faqat o'zgartirilgan va ularga bog'liq bo'lgan modullargina qayta kompilyatsiya qilinadi. Oddiy dasturlar uchun shunga teng kuchli *Alt* + *F9* bosiladi.

Ctrl + *F9* — tahrirlashning faol oynasidagi dasturni bajarishga yuborish.

Dasturlarni tuzatish klavishlari:

Alt + *F5* — dastur bajarilishi natijalarini ko'rish.

F8 — dasturning qadam-baqadam bajarilishi. Protsedura va funksiyalarni chaqirish bitta operator (bitta qadam) kabi bajariladi.

F7 — dasturni qadam-baqadam bajarish. Protsedura yoki funksiyalarni chaqirishda uning matniga kirish va operatorlarni qadam-baqadam bajarish sodir bo'ladi.

F4 — dastur qadam-baqadam bajarilishining joriy satridan kursor joylashgan satrgacha dastur qismining bajarilishi.

Ctrl + *F2* — dasturni tahrir qilish ishini yakunlaydi va uni xotiradan bo'shatadi.

Ctrl + *F3* — *Call Stack* oynasini ochadi. Bu oynada shu daqiqada bajarilayotgan protseduragacha chiqarilgan dastur protseduralari ro'yxati ketma-ketligi ko'rsatiladi.

Ctrl + *F4* — *Evaluate and modify* oynasini ochadi. Bu oynada qiymati aniqlanishi talab etilgan ifodani ko'rsatish, dastur o'zgaruvchilari qiymatlarini va berilganlar elementlarini qarab chiqish hamda ularni o'zgartirish kerak.

Ctrl + *F7* — *Add Watch* muloqot oynani ochadi. Bu oynaga dasturchi tuzatishni bajarish vaqtida, qiymatlari qiziqtiradigan ifoda yoki o'zgaruvchi nomini yozishi mumkin.



2.1. QUYI MURAKKABLIKDAGI MASALALAR



2.1.1. O'zgaruvchilarni e'lon qilish

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni esda tutish kerak:

- dasturning har bir o'zgaruvchisi e'lon qilinishi kerak;
- o'zgaruvchilar *var* so'zi bilan boshlanadigan bo'limda e'lon qilinadi;
- o'zgaruvchilarni e'lon qilish ko'rsatmasi quyidagi ko'rinishga ega:

O'zgaruvchi Ismi: Turi;

- o'zgaruvchi nomi sifatida lotin alifbosining harflari va raqam (birinchi belgi harf bo'lishi kerak) ishlatiladi;
- *Turbo Pascal* tilining asosiy sonli turlari quyidagilar: *integer* (butun), *real* (haqiqiy kasr);
- o'zgaruvchi e'lon qilingan ko'rsatmadan keyin o'zgaruvchining qo'llanilishini ko'rsatish tavsiya qilinadi.



MASALALAR

1. $y=x^2$ funksiyaning qiymatini hisoblash uchun zarur o'zgaruvchilarni e'lon qiling.
2. Funt o'lchovidagi og'irlikni kilogrammga aylantirish uchun zaruriy o'zgaruvchilarni e'lon qiling (bir funt = 409,5 g).
3. Parallelepiped hajmini hisoblash uchun zaruriy o'zgaruvchilarni e'lon qiling.
4. Doira yuzasini hisoblash uchun zaruriy o'zgaruvchilarni e'lon qiling ($S=\pi \cdot r^2$).
5. Silindr sirt yuzasi va hajmini hisoblash uchun zaruriy o'zgaruvchilarni e'lon qiling ($v=2 \cdot \pi r(h+r)$; $v=\pi \cdot R^2 \cdot h$).
6. Bir necha daftarlar, qalamlar va chizg'ichdan iborat xaridning qiymatini hisoblash uchun zaruriy o'zgaruvchilarni e'lon qiling.



2.1.2. O'zlashtirish ko'rsatmasi*

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni esda tutish kerak:

- o'zlashtirish ko'rsatmasi o'zgaruvchilar qiymatini o'zgartirish, shuningdek, formula bo'yicha hisoblashda ishlatiladi;
- o'zlashtirish ko'rsatmasining o'ng tomonidagi ifoda turi chap tomondagi o'zgaruvchi turiga mos kelishi kerak (aks holda kompilyator — *Type Miss match* — turlar mos kelmaydi, degan axborot chiqaradi).



MASALALAR

- O'zgaruvchiga 1,5 qiymatni beruvchi ko'rsatmani yozing.
- Summa* o'zgaruvchisiga nol qiymat beruvchi ko'rsatmani yozing.
- n o'zgaruvchi qiymatini 1 ga oshiruvchi ko'rsatmani yozing.
- counter* o'zgaruvchi qiymatini 2 ga kamaytiruvchi ko'rsatmani yozing.
- $X1$ va $X2$ o'zgaruvchilar qiymatining o'rta arifmetigini topish ko'rsatmasini yozing.
- $y = -2,7x^3 + 0,23x^2 - 14$ funksiya qiymatini hisoblash ko'rsatmasini yozing.
- Masofani kilometrdan verstlarga o'tkazish formulasi bo'yicha hisoblash ko'rsatmasini yozing (bir verst = 1066,8 m).
- To'rtburchak yuzini hisoblash ko'rsatmasini yozing.
- Uchburchak yuzini hisoblash ko'rsatmasini yozing.
- Shar sirt yuzasi va hajmini hisoblash ko'rsatmasini yozing:

$$(v = \frac{3}{4} \pi R^3; S = 4\pi R^2).$$

- G'ovak silindr hajmini hisoblash ko'rsatmasini yozing:

$$(v = \pi \cdot R/R_1^2 - R_2^2).$$

- Konus hajmini hisoblash ko'rsatmasini yozing:

$$(S = \frac{1}{3} \cdot \pi \cdot R^2 \cdot h).$$



2.1.3. Chiqarish ko'rsatmasi

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslang:

* Ba'zi bir adabiyotlarda «ko'rsatma» o'rnida «operator» so'zi ishlatiladi.

- *write* va *writeln* ko'rsatmalari monitor ekraniga axborotlar va o'zgaruvchi qiymatlarini chiqarish uchun ishlatiladi;
- bitta *write* (*writeln*) ko'rsatmasi ekranga bir necha o'zgaruvchi qiymatlarini va (yoki) bir necha axborotlarni chiqarishi mumkin;
- parametrsiz *writeln* ko'rsatmasi kursorni ekraning navbatdagi satr boshiga o'tkazadi.



MASALALAR

19. Ekranga ism va familiyani chiqaruvchi dasturni yozing.
20. Ekranga quyidagi to'rtlikni chiqaruvchi dasturni yozing:

Inson aqli — galaktika yo'lida chiroq,
 Balki kosmos mozorida ko'milar tanim?
 Yo'q-yo'q, inson darichasin ochgan u chog'da,
 Tarixlarga — u, o'lmas, deb yozdi, Vatanim.

Toshpo'lat Hamid

21. Kasr qismida uchta raqam bo'lgan o'zgaruvchi qiymatini ekranga chiqarish ko'rsatmasini yozing.

22. Quyidagi ko'rsatmalar nima qilishini aniqlang:

write (*a*);

write ('*a*').

23. *a*, *b* va *c* kasr o'zgaruvchi qiymatlarini bir satrga chiqaruvchi ko'rsatmani yozing. Har bir son o'nlik nuqtadan keyin ikki xona aniqlik bilan chiqarilishi kerak (*writeln* (*a:6:2*, '*...*'; *b:6:2*, '*...*'; *c:6:2*)).

24. *a*, *b* va *c* kasr o'zgaruvchi qiymatlarining har birini alohida satrdan o'zgaruvchi nomlari ko'rsatilgan ko'rinishda chiqaring.

Qo'shimcha ko'rsatmalar

Crt standart kutubxona *write* va *writeln* ko'rsatmalari bilan chiqariladigan o'zgaruvchilarga tag rang va belgi ranglarini bera oluvchi protseduralarga ega. Bu bo'lim masalalarini yechishga o'tishdan oldin, quyidagilarni eslash kerak:

- *Crt* kutubxona protsedura va funksiyalarini ishlata olish uchun dasturdagi birinchi ko'rsatma *Uses Crt* bo'lishi kerak;
- belgilar rangini *TextBackGround* protsedurasi o'rnatadi;
- tag rangini *TextColor* protsedura belgilaydi;
- *TextBackGround* va *TextColor* protseduralarini ishlatishda rang tanlangan o'zgaruvchi yoki butun o'zgaruvchilar bilan berilishi mumkin;
- *Clr Scr* protsedura ekranni tozalaydi;

- dastur matniga *ClrScr*, *TextBackGround* va *TextColor* protseduralaridan foydalanish mumkin bo'lishi uchun *Uses Crt* direktivasini kiritish kerak.



MASALALAR

25. Quyidagi to'rtlikni kulrang harflar bilan ko'k rangli yuzaga chiqarish dasturini yozing:

Dunyo bamisoli bir zarrin ko'za,
Suvi goh achchig'-u, goh shirin bo'za.
Ey, inson, umringga bino qo'yma ko'p,
Qazoyi muallaq turar bosh uzra!

Pahlavon Mahmud

26. Quyidagi kamalak rangini bildiruvchi so'zlar ketma-ketligini ekranga chiqarish dasturini yozing: qizil, olovrang, sariq, yashil, havorang, ko'k, binafsharang. Har bir rang nomini bildiruvchi so'z shu rangli harflar bilan yozilishi kerak.



2.1.4. Kiritish ko'rsatmasi

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- Dastur bilan ishlash vaqtida boshlang'ich qiymatlarni klaviaturadan kiritish uchun *readln* ko'rsatmasi berilgan.
- Bitta *readln* ko'rsatmasi bilan bir nechta o'zgaruvchilarni kiritish mumkin.
- Dastur ishi vaqtida kiritiladigan ma'lumotlar turi *readln* ko'rsatmasidagi o'zgaruvchi turiga mos kelishi kerak.
- O'zgaruvchining klaviaturadan kiritilayotgan qiymati kiritilgan ma'lumotlar turiga mos kelmasa, dastur ishini tugatadi va ekranga *Error 106 : Invalid numeric format* (agar, dastur *Turbo Pascal* muhitidan ishga tushirilgan bo'lsa) yoki *Run time error 106* (agar dastur operatsion tizimdan ishga tushirilgan bo'lsa) degan axborot chiqadi.



MASALALAR

27. *U* va *r* o'zgaruvchilar qiymatini kiritishni ta'minlovchi ko'rsatma yozing. Dastur ishlash vaqtida dasturchi har bir sonni terganidan keyin *<Enter>* klavishini bosadi, deb hisoblanadi. Foydalanuvchi sonlarni har xil satrdan kiritadi.

28. *U* va *r* o'zgaruvchilar qiymatini kiritishni ta'minlovchi ko'rsatma yozing. Foydalanuvchi sonlarni bir satrdan kiritadi.

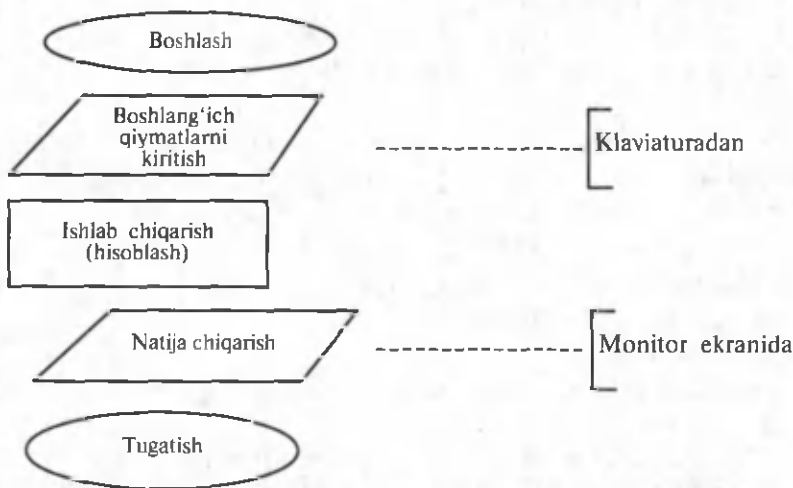
29. Zarur o'zgaruvchilarni e'lon qiling va boshlang'ich ma'lumotlarning kiritilishini, silindr hajmini hisoblashni ta'minlaydigan dastur bo'lagini yozing.



2.1.5. Chiziqli tasnifli dasturlar

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- chiziqli tasnifli dasturlar eng oddiy dasturlardir va ular, odatda, formula bo'yicha hisoblashlarni amalga oshirishda ishlatiladi;
- chiziqli tasnifli dasturlarda ko'rsatmalar bir-biridan keyin chiziqli ketma-ketlikda bajarib boriladi;
- chiziqli tasnifli dastur algoritmi quyidagicha bayon etilishi mumkin:



! *Izoh:* masala shartlarida dasturchi kiritadigan sonlar yo'g'on shrift bilan yozilgan.



MASALALAR

30. Parallelogramm yuzasini hisoblash dasturini tuzing. Ekranga natijalarning quyidagicha chiqarilishi tavsiya etiladi:

Parallelogramm yuzasini hisoblash.

Boshlang'ich qiymatlarni kiriting:

Uzunlik (sm) → 9

Kenglik (sm) → 7.5

Parallelogramm yuzi: 67.50 sm².

31. Parallelepiped sirt yuzasini hisoblash dasturini tuzing. Quyida dastur ishlash vaqtidagi ekranning tavsiya qilinadigan ko'rinishi berilgan.

Parallelepiped sirt yuzasini hisoblash.

Boshlang'ich qiymatlarni kiriting:

Uzunlik (sm) \rightarrow 9

Kenglik (sm) \rightarrow 7.5

Balandligi (sm) \rightarrow 5

Sirt yuzasi: 90.00 sm^2 .

32. Silindr hajmini hisoblash dasturini yozing. Quyida dastur ishlash vaqtidagi ekranning tavsiya qilinadigan ko'rinishi berilgan.

Silindr hajmini hisoblash.

Boshlang'ich qiymatlarni kiriting:

Asos radiusi (sm) \rightarrow 5

Silindr balandligi (sm) \rightarrow 10

Dastur ishini tugatish uchun <Enter>ni bosing.

Silindr hajmi: 1570.80 sm^3 .

33. Bir necha daftar va qalamlardan iborat xarid qiymatini hisoblash dasturini yozing. Quyida dastur ishlash vaqtidagi tavsiya qilinadigan ekran ko'rinishi berilgan..

Xarid qiymatini hisoblash.

Boshlang'ich qiymatlarni kiriting:

Daftar bahosi (so'm) \rightarrow 40

Daftarlar soni \rightarrow 5

Qalam bahosi (so'm) \rightarrow 30

Qalam soni \rightarrow 2

Xarid qiymati: 260 so'm .

34. Agar berilgan uchburchakning ikki tomoni va ular o'rtasidagi burchagi ma'lum bo'lsa, uchburchak yuzini hisoblash dasturini yozing. Quyida dastur ishlash vaqtidagi tavsiya qilinadigan ekran ko'rinishi berilgan.

Uchburchak yuzini hisoblash.

Uchburchak 2 tomoni uzunligi: (sm) \rightarrow 25 17

Uchburchak tomonlari orasidagi burchak kattaligi (gradus) \rightarrow 30

Uchburchak yuzi: 106.25 sm^2 .

35. Ikkita parallel ulangan qarshiliklardan iborat elektr zanjir qarshiligini hisoblash dasturini yozing. Quyida dastur ishlash vaqtidagi tavsiya qilinadigan ekran ko'rinishi berilgan:

Elementlari parallel ulangan elektr zanjir qarshiligini hisoblash.

Boshlang'ich qiymatlarni kiriting:

Birinci qarshilik kattaligi (Om) → 15

Ikkinchi qarshilik kattaligi (Om) → 20

Zanjir qarshiligi: 8.57 Om.

36. Avtomobilda dala hovliga borish va qaytish qiymatini hisoblash dasturini yozing. Quyidagilar boshlang'ich qiymatlar: dala hovligacha masofa (km.larda); 100 km masofaga ketadigan benzin miqdori, 1 litr benzin bahosi. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Dala hovliga borish va qaytish safari qiymati.

Dala hovligacha masofa (km) → 67

Benzin sarfi (100 km masofaga, litr) → 8.5

Litr benzin bahosi (so'm) → 150

Dala hovliga borish va qaytish qiymati: 1708.5 so'm.

37. Silindr sirt yuzasini hisoblash dasturini yozing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Silindr sirt yuzasini hisoblash.

Boshlang'ich qiymatlar:

Asos radiusi (sm) → 5,5

Silindr balandligi (sm) → 7

Silindr sirt yuzasi: 431.97 sm².

38. Ixtiyoriy berilgan qiymatlarda parallelepiped hajmini hisoblash dasturini yozing.

39. Ixtiyoriy berilgan qiymatlarda kub hajmini hisoblash dasturini yozing.

40. Ixtiyoriy berilgan qiymatlarda uchburchak yuzini hisoblash dasturini yozing.

41. Verstlarda berilgan masofa uzunligini km.larga aylantirish dasturini tuzing (1 verst = 1066.8 m). Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Masofani verstlardan km.ga aylantirish.

Verstlardagi masofani kiriting va <Enter>ni bosing → 100

100 verst → bu 106.68 km.

42. Kartada ko'rsatilgan ikkita aholi punkti o'rtasidagi masofani hisoblash dasturini yozing (karta masshtabi 120 km bir sm.da. Kartadagi ikki nuqta orasidagi masofa 3.5 sm).

43. Omonat mablag'lar bo'yicha daromad miqdorini hisoblash dasturini tuzing. Daromad foizi (yillik foizga nisbatan) va omonatni saqlash vaqti (kunlarda) omonat miqdori dastur ishlash vaqtida beriladi. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Omonat bo'yicha daromadni hisoblash.

Boshlang'ich qiymatlarni kiriting:

Omonat miqdori (so'm) → 2500

Omonat muddati (kunlarda) → 30

Qo'shimcha foiz (yillik) → 20

Daromad: 41.10 so'm.

Omonat muddati oxiridagi yig'indi: 2541.10 so'm.

44. Minutlarda berilgan vaqt oralig'i miqdorini soat va minutlarga aylantirish dasturini yozing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Vaqt oralig'i miqdorini kiriting (minutlarda) va <Enter> ni bosing → 150

150 minut → bu 2 soat 30 minut.

45. Klaviaturadan kiritilgan kasr sonni pul formatiga o'tkazish dasturini yozing. Masalan, 12.5 soni 12 so'm 50 tiyin ko'rinishiga keltirilishi kerak.

Sonni pul formatiga o'zgartirish.

Kasr sonni kiriting → 23.6

23.6 so'm → bu 23 so'm 60 tiyin.



2.1.6. Tanlov (IF) ko'rsatmasi

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- *if* ko'rsatma dasturning keyingi yo'lida (bajarilishi kerak bo'lgan ko'rsatmalar ketma-ketligida) ikki yo'nalishdan birini tanlash uchun ishlatiladi;

- ko'rsatmalar ketma-ketligini tanlash shartning bajarilishiga bog'liq ravishda dasturning bajarilish vaqtida amalga oshiriladi;

- shart-mantiqiy turdagi ifoda bo'lib, u ikkita qiymatdan birini tanlashi mumkin: *True* (haqiqat — shart bajariladi) yoki *False* (yolg'on — shart bajarilmaydi);

- oddiy mantiqiy *and* (mantiqiy «VA») va *or* (mantiqiy «YOKI») amallari yordamida oddiy shartlardan murakkablarini tuzish mumkin.



Izoh: masala shartlarida dasturchi kiritadigan sonlar yo'g'on shrift bilan yozilgan.



MASALALAR

46. Ikki sonni bo'lishdan hosil bo'ladigan bo'linmani hisoblash dasturini tuzing. Dastur foydalanuvchi kiritgan qiymatlarning

to'g'riligini tekshirishi kerak, agar ular noto'g'ri bo'lsa (bo'luvchi nolga teng), xato to'g'risidagi axborot chiqarilsin. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Bo'linmani hisoblash.

Bitta satrdan bo'linuvchi va bo'luvchini kiriting.

Keyin <Enter>ni bosing → 12 0

Adashdingiz. Bo'luvchi nolga teng bo'lmasligi kerak.

47. Ichi teshik halqa yuzasini hisoblash dasturini tuzing. Dastur boshlang'ich qiymatlarning to'g'riligini tekshirishi kerak. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Halqa yuzini hisoblash.

Boshlang'ich qiymatlarni kiriting:

Halqa radiusi (sm) → 3.5

Teshik radiusi (sm) → 7

Xato! Teshik radiusi halqa radiusidan katta bo'lmasligi kerak!

48. Ikki qarshilikdan iborat elektr zanjir qarshiligini hisoblash dasturini tuzing. Qarshiliklar ketma-ket yoki parallel ulangan bo'lishi mumkin. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Elektr zanjir qarshiligini hisoblash.

Boshlang'ich qiymatlarni kiriting:

Birinci qarshilik miqdori (Om) → 15

Ikkinchi qarshilik miqdori (Om) → 27.3

Ulanish turi (1-ketma-ket, 2-parallel) → 2

Zanjir qarshiligi: 9.68 Om.

49. Kvadrat tenglamani yechish dasturini yozing. Dastur boshlang'ich qiymatlarning to'g'riligini tekshirishi kerak. Ikkinchi darajali noma'lum son oldidagi koeffitsient nolga teng bo'lsa, tegishli axborot chiqishi kerak. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Kvadrat tenglamani yechish.

Bitta satrdan koeffitsientlar qiymatini kiriting va <Enter>ni bosing → 12 27 - 10

Tenglama ildizlari:

$$x1 = - 25.551$$

$$x2 = - 28.449$$

50. Xaridorning imtiyozini hisobga olgan holda xarid qiymatini hisoblash dasturini tuzing. Agar xarid yig'indisi 5000 so'mdan katta bo'lsa 3 %, 10000 so'mdan katta bo'lsa 5 % dan xaridorga yon bosiladi. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Imtiyozni hisobga olgan holda xarid qiymatini hisoblash.

Xarid qiymatini kiriting va <Enter>ni bosing → 6400

Sizga 3 % imtiyoz beriladi.

Imtiyoz hisobga olingan holda xarid jami: 6208.0 so'm.

51. Xarid jami 10000 so'mdan ko'p bo'lsa, xarid qiymatiga 10 % yon bosiladi. Shuni hisobga oluvchi xarid qiymatini hisoblash dasturini yozing.

52. Buyuk o'zbek shoiri Alisher Navoiy tug'ilgan yili to'g'riligini tekshirish dasturini tuzing. Xato javob berilganda, dastur to'g'ri javobni chiqarsin. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Alisher Navoiy qachon tug'ilgan?

Sanani kiriting va <Enter>ni bosing → 1563

Adashdingiz. Alisher Navoiy 1441-yilda tug'ilgan.

53. Fanlar tarixini bilish bo'yicha bilimni tekshirish dasturini tuzing. Dastur savolni va unga uchta javobni chiqarishi kerak. Foydalanuvchi to'g'ri javobni tanlashi va uning raqamini kiritishi kerak. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Kibernetika fani asoschisi:

1. Andre Mari Amper.

2. Isaak Nyuton.

3. Norbert Vinner.

To'g'ri javob raqamini kiriting va <Enter>ni bosing → 2

Adashdingiz. Kibernetika fani asoschisi — Norbert Vinner → .

54. Klaviaturadan kiritilgan ikkita butun sonni taqqoslash dasturini tuzing. Dastur qaysi son kattaligini ko'rsatishi va agar sonlar teng bo'lsa, tegishli axborotni chiqarishi kerak. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Bitta satrdan ikkita butun sonni kiriting va <Enter> ni bosing
→ 34 67

34 soni 67 dan kichik.

55. Ko'paytirilishi kerak bo'lgan ikkita bir xil xonali sonlarni chiqaruvchi, foydalanuvchi javobini so'rovchi, uni tekshiruvchi va «To'g'ri!» yoki «Adashdingiz» axborotlarini va to'g'ri javobni chiqaruvchi dastur tuzing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

6 x 7 nechaga teng?

Javobni kiriting va <Enter>ni bosing → 56

Adashdingiz. $6 \times 7 = 42$.

56. Ikkinchi jahon urushi boshlanishi sanasini bilishning tekshiruvchi dasturini tuzing. Dastur to'g'ri javobni baholashi, xato javobda u to'g'risini chiqarishi kerak.

57. Biror soha bo'yicha bilimni tekshiruvchi dastur tuzing. Dastur savolni va unga uch xil javob variantini chiqarishi kerak. Foydalanuvchi to'g'ri javob raqamini kiritishi kerak. To'g'ri va xato javoblar dastur tomonidan izohlanishi kerak.

58. 100 gacha bo'lgan sonlar bilan ayirish amalini bajarishga misol chiqaruvchi, foydalanuvchi javobini tekshiruvchi, unga izohlar beruvchi dastur tuzing. Quyida tavsiya etiladigan ekran ko'rinishi beriladi:

83—17 necha bo'ladi?

Javobni kiriting va <Enter>ni bosing → 67

Adashdingiz. $83 - 17 = 66$.

59. Kiritilgan sonning juftligini tekshiruvchi dastur yozing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Butun sonni kiriting va <Enter>ni bosing → 23

23 soni — toq.

60. Klaviaturadan kiritilgan son uchga bo'linishini tekshiruvchi dastur tuzing. Javoblar izohli bo'lsin. Masalan,

Butun sonni kiriting va <Enter>ni bosing → 451

451 soni uchga bo'linmaydi.

61. Shanba va yakshanba kungi telefon orqali so'zlashuvlarning 20 % arzonligini hisobga olgan holda so'zlashuvlar qiymatini aniqlovchi dastur tuzing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Telefon orqali so'zlashuvlar qiymati.

Boshlang'ich qiymatlarni kiriting:

So'zlashuv vaqti (butun son. Minut) → 3

Hafta kuni (1-dushanba, ..., 7-yakshanba) → 6

20 % arzonlashtiriladi.

So'zlashuvlar qiymati: 55.2 so'm.

62. Foydalanuvchining optimal vaznini hisoblovchi, uni real vazn bilan taqqoslovchi va foydalanuvchiga semirish yoki ozish zaruriyati to'g'risida tavsiyalar beruvchi dastur tuzing. Optimal vazn bo'y uzunligidan (sm) 100 ni ayirish natijasiga teng. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Bitta satrdan bo'shliq (probel) orqali bo'y (sm) va vazn (kg) qiymatlarini kiriting va <Enter>ni bosing → 170 68

Siz 2 kg semirishingiz kerak.

63. 1 dan 100 gacha bo'lgan sonlarning juftlarini kvadratga, toqlarini kub darajaga ko'tarib, ularning yig'indisini hisoblang.



2.1.7. CASE ko'rsatmasi

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- Case ko'rsatma dastur keyingi yo'lining (bajarilishi kerak bo'ladigan ko'rsatmalar ketma-ketligida) bir nechta yo'nalishlaridan birini tanlash kerak bo'lganda ishlatiladi.

- Ketma-ket ko'rsatmalardan birini tanlash dastur bajarilishi vaqtida, o'zgaruvchining ko'rsatmalar guruhi boshida berilgan saralovchi o'zgarmas qiymatiga tengligiga bog'liq ravishda amalga oshiriladi.

- Saralovchi o'zgaruvchi sifatida butun (*integer*) yoki ramziy (*char*) turdagi o'zgaruvchilardan foydalaniladi.



MASALALAR

64. Foydalanuvchidan oy raqamini so'rovchi va bu oy yilning qaysi mavsumiga tegishli ekanligi to'g'risida javob beruvchi dastur tuzing. Agar foydalanuvchi mumkin bo'lmagan son kiritrsa, dastur xato son berilganligi to'g'risida axborot chiqarsin. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Oy raqamini kiriting (1 dan 12 gacha) va `<Enter>`ni bosing → 11 Qish.

65. Shaharlararo telefon orqali so'zlashuv qiymatini hisoblovchi dastur tuzing (u minut bahosi — abonentning shahargacha bo'lgan masofasi — uzunligi bilan aniqlanadi). Dastur uchun shahar kodi va so'zlashuv vaqti boshlang'ich qiymatlar bo'lib xizmat qiladi. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Telefon orqali so'zlashuvni hisoblash.

Boshlang'ich qiymatlarni kiriting:

Shahar kodi → 371

So'zlashuv vaqti (butun son minut) → 3

Shahar: Toshkent.

Minut bahosi: 150 so'm.

So'zlashuv bahosi: 450 so'm.

Shahar	Kod	Bir minut bahosi, so'm
Toshkent	371	150
Samarqand	366	120
Qarshi	375	100
Buxoro	365	130

66. Keyingi kun sanasini hisoblovchi dastur tuzing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Bugungi sanani raqamlarda kiringing (son oy yil) → 31. 12. 2005.

Oyning oxirgi kuni!

Kelayotgan Yangi yil bilan!

Ertaga 01.01.2006.

67. Foydalanuvchidan hafta kuni raqamini so'rovchi va shu kunning nomlanishini ekranga chiqaruvchi dastur tuzing.



2.1.8. FOR ko'rsatmasi

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- *for* ko'rsatma dasturda, ishlab chiqish vaqtida aniqlanuvchi, takrorlashlarning ma'lum sondagi sikllarini tashkil etish uchun ishlatiladi.

- Takrorlashlar soni o'zgaruvchi — hisobchining boshlang'ich va oxirgi qiymatlari bilan aniqlanadi.

- O'zgaruvchi — hisobchi butun turda (*integer*) bo'lishi kerak.



Izoh: masala shartlarida dasturchi kiritadigan sonlar yo'g'on shrift bilan yozilgan.



MASALALAR

68. Birinchi o'nta butun musbat sonlar kvadratlari jadvalini chiqaruvchi dastur tuzing. Quyida dastur ishlash vaqtida hosil bo'lishi kerak bo'ladigan ekran ko'rinishi tavsiya etiladi:

Kvadratlar jadvali

Son	Kvadrat
1	1
2	4
3	9
4	16
5	25

Son	Kvadrat
6	36
7	49
8	64
9	81
10	100

69. Birinchi «*n*» ta butun musbat sonlar yig'indisini hisoblovchi dasturni yozing. Qo'shiladigan sonlar dastur ishlayotgan vaqtda kiritilishi kerak. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Musbat sonlar yig'indisini hisoblash.

Qo'shiladigan sonlar miqdorini kiriting $\rightarrow 20$

Birinchi 20 ta musbat sonlar yig'indisi 210 ga teng.

70. Birinchi butun musbat toq sonlar kvadratlari jadvalini chiqaruvchi dastur tuzing.

71. Dastlabki n ta butun musbat sonlar yig'indisini hisoblovchi dastur tuzing.

72. 1,3,5,7, ... qatorning birinchi n ta hadi yig'indisini hisoblash dasturini tuzing. Qo'shiluvchi hadlar soni dastur ishi vaqtida beriladi. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

1,3,5,7, ... qatorning xususiy yig'indisini topish.

Qatorning qo'shiluvchi hadlar sonini kiriting $\rightarrow 15$

Qatorning dastlabki 15 ta hadi yig'indisi: 330.

73. $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$ qatorning birinchi n ta hadi yig'indisini hisoblash dasturini tuzing. Qo'shiluvchi hadlar soni dastur ishi vaqtida beriladi. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$ qator xususiy yig'indisini hisoblash.

Qator qo'shiluvchi hadlar sonini kiriting $\rightarrow 15$

Qatorning dastlabki 15 ta hadi yig'indisi: 3.3182.

74. Ikki sonining 0 dan 10 gacha bo'lgan darajalari jadvalini chiqaruvchi dastur tuzing. Quyida dasturning ish vaqtida tavsiya etiladigan ekran ko'rinishi keltiriladi:

Ikkining darajalari	Natija
0	1
1	2
2	4
3	8
4	16
5	32

Ikkining darajalari	Natija
6	64
7	128
8	256
9	512
10	1024

75. Sonning faktorialini hisoblovchi dastur tuzing.

76. Klaviaturadan kiritilgan 5 ta kasr sonlarning o'rtqa qiymatini hisoblovchi dastur tuzing.

77. Klaviaturadan kiritilgan 1 dan 10 gacha bo'lgan sonlarning takrorlanishini ham hisobga olib, ularning o'rtqa arifmetik qiymatini hisoblovchi dastur tuzing.

78. Mahsulotning og'irligiga ko'ra, bahoni ekranga chiqaruvchi jadval tuzing. Masalan, 1 kg 16 so'm 50 tiyin bo'lsin.

79. $y = -2,4x^2 + 5x - 3$ funksiyaning -2 dan 2 gacha sohada $0,5$ qadam bilan o'zgarandagi qiymatlari jadvalini chiqaruvchi dasturni tuzing. Quyida dasturning ish vaqtida tavsiya etiladigan ekran ko'rinishi keltiriladi:

x	y
-2	-22,60
-1,5	-15,90
-1	-10,40
-0,5	-6,10
0	-3,00

x	y
0,5	-1,10
1	0,40
1,5	-0,90
2	-2,60

80. $y = |x + 2|$ funksiyaning qiymatlar jadvalini ekranga chiqaruvchi dastur tuzing. « ∞ » -4 dan 4 gacha $0,5$ qadam bilan o'zgaradi.

81. $y = |x - 2| + |x + 1|$ funksiyaning qiymatlar jadvalini ekranga chiqaruvchi dastur tuzing. « ∞ » -4 dan 4 gacha $0,5$ qadam bilan o'zgaradi.

82. Klaviaturadan beshta kasr sondan iborat ketma-ketlikni kiritish dasturini tuzing. Dasturda har bir son kiritilgandan keyin ketma-ketlikning hosil bo'lgan qismi uchun o'rtqa arifmetik qiymat chiqarilishi hisobga olingan bo'lsin. Dasturning ish vaqtida tavsiya etiladigan ekran ko'rinishi quyida keltirilgan.

Kasr sonlar ketma-ketligini qayta ishlash.

Har bir son kiritilgandan keyin <Enter>ni bosib \rightarrow 12.3

Kiritilgan son: 1 ta yig'indi 12.30 o'rtqa arifmetik: 12.30 \rightarrow 15

Kiritilgan son: 2 ta yig'indi 27.30 o'rtqa arifmetik: 13.65 \rightarrow 10

Kiritilgan son: 3 ta yig'indi 37.30 o'rtqa arifmetik: 12.43 \rightarrow 5.6

Kiritilgan son: 4 ta yig'indi 42.90 o'rtqa arifmetik: 10.73 \rightarrow 11.5

Kiritilgan son: 5 ta yig'indi 54.40 o'rtqa arifmetik: 10.88

Dastur ishini tugatish uchun <Enter>ni bosib.

83. Klaviaturadan kiritilayotgan kasr sonlar ketma-ketligining o'rtta arifmetigini hisoblash dasturini tuzing. Oxirgi son kiritilganidan keyin dastur ketma-ketligining eng kichik va eng katta sonini chiqarishi kerak. Sonlar miqdori dastur ishi vaqtida beriladi. Quyida dasturning ishi vaqtida tavsiya etiladigan ekran ko'rinishi keltirilgan.

Kasr sonlar ketma-ketligini qayta ishlash.

Ketma-ketlikdagi sonlar miqdorini kiriting → 5

Ketma-ketlikni kiriting. Har bir son kiritilgandan keyin

<Enter>ni bosing.

→ 5.4

→ 7.8

→ 3.0

→ 1.5

→ 2.3

Sonlar miqdori: 5

O'rtta arifmetik: 4.00

Minimal son: 1.5

Maksimal son: 7.8

Dasturning ishini tugatish uchun <Enter> klavishini bosing.

84. 1 dan 10 gacha sohadagi 10 ta tasodifiy sondan iborat uchta ketma-ketlikni ajratuvchi dasturni yozing. Dastur har bir ketma-ketlikni ekranga chiqaradi va har bir ketma-ketlik o'rtta arifmetigini hisoblaydi. Dasturning ishlash vaqtidagi ekran ko'rinishi quyida keltirilgan:

*** Tasodifiy sonlar ***

6	10	4	2	5	8	1	7	7	3	O'rtta arifmetigi 5.30
10	3	6	1	10	1	3	8	7	6	O'rtta arifmetigi 5.50
5	2	2	5	4	2	2	1	6	10	O'rtta arifmetigi 3.90

Ishni tugatish uchun <Enter> klavishini bosing.

85. $y=|x|$ funksiyaning argument sohasi -4 dan 4 gacha $0,5$ qadam bilan o'zgarandagi qiymatlari jadvalini hosil qilish dasturini tuzing.

86. Ekranga ko'paytirish (masalan, 7 ga) jadvalini chiqarish dasturini tuzing. Dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi quyidagicha:

$7 \times 2 = 14$	$7 \times 6 = 42$
$7 \times 3 = 21$	$7 \times 7 = 49$
$7 \times 4 = 28$	$7 \times 8 = 56$
$7 \times 5 = 35$	$7 \times 9 = 63$

87. Klaviaturadan 0 dan 255 gacha bo'lgan sohadagi o'nlik butun sonni ikkilikda berish dasturini yozing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

O'nlik sonni ikkilikka aylantirish.

[0; 255] sohadan butun son kiriting va <Enter>ni bosing → 164

O'nlik 164 soniga 10100100 ikkilik soni mos keladi.

Dastur ishini tugatish uchun <Enter> klavishini bosing.

88. Ekraniga Pifagorning ko'paytirish jadvalini chiqarish dasturini tuzing. Dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi quyidagicha:

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90

89. $1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots$ qatorning xususiy yig'indisini hisoblaydigan va hosil bo'lgan qiymatni $\pi/4$ bilan taqqoslaydigan (yetarli darajada katta miqdordagi hadlar qo'shilganda, xususiy yig'indi $\pi/4$ ga yaqinlashib boradi) dastur tuzing.

90. Ko'paytirish jadvalini bilishni tekshiruvchi dastur tuzing. Dastur bo'yicha 10 ta misol ishlanishi, 10 ta to'g'ri javob uchun «a'lo», 9 va 8 tasi uchun «yaxshi», 7 tasi uchun «qoniqarli», 6 va undan kam to'g'ri javoblar uchun «yomon» baho qo'yilishi kerak. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan:

$$5 \times 3 = 15$$

$$7 \times 7 = 49$$

$$1 \times 4 = 4$$

$$4 \times 3 = 12$$

$$9 \times 4 = 36$$

$$8 \times 8 = 64$$

$$7 \times 8 = 52$$

$$\text{Adashdingiz. } 7 \times 8 = 56.$$

$$4 \times 7 = 28$$

$$3 \times 5 = 15$$

$$2 \times 5 = 10$$

To'g'ri javoblar: 9.

Baho: yaxshi.

91. 100 gacha bo'lgan sonlarni qo'shish va ayirishni bajarishni tekshirish. Dastur bo'yicha 10 ta misol ishlanishi kerak, har bir misolda kamayuvchi son ayiriluvchi sondan katta yoki unga teng bo'lishi kerak (manfiy son hosil qilish mumkin emas). Baholash quyidagicha bo'ladi: 10 ta to'g'ri javob uchun «a'lo», 9 va 8 tasi uchun «yaxshi», 7 tasi uchun «qoniqarli», 6 va undan kam to'g'ri javoblar uchun «yomon» baho. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Sonlarni qo'shish va ayirish ko'nikmasini tekshirish.

Misoldan keyin javobni kiriting va <Enter>ni bosing.

$$75 - 4 = 71$$

$$35 - 9 = 29$$

Adashdingiz. $35 - 9 = 26$.

$$14 - 1 = 13$$

$$6 - 5 = 1$$

$$37 - 19 = 28$$

Adashdingiz. $37 - 19 = 18$.

$$53 - 14 = 39$$

$$94 - 87 = 7$$

$$90 - 16 = 74$$

$$4 - 2 = 2$$

$$89 - 41 = 48$$

To'g'ri javoblar: 8.

Baho: yaxshi.

92. Ekranga, masalan, uch minut mobaynida yoki foydalanuvchi ixtiyoriy klavishni bosgunga qadar ishlaydigan «elektron soat»ni chiqaruvchi dasturni tuzing.

93. Ekranga shaxmat taxtasi ifodasini chiqaruvchi dasturni tuzing. Yulduzchalar (*) qora kataklarni, bo'shliq oq kataklarni bildirsin.

94. «m» va «n» o'zgaruvchilari oralig'ida qiymatlar kvadratlarini yig'indisini hisoblang.

95. $a < 20$; $1 < b < 20$ bo'lganda $a^2 + b^2 = c^2$ tenglikni qanoatlantiruvchi (Pifagor) sonlarini aniqlang.

96. Raqamlar kublarining yig'indisi sonning o'ziga teng bo'ladigan barcha uch xonali sonlarni toping (masalan, $407 = 4^3 + 0^3 + 7^3 = 64 + 0 + 343$).



2.1.9. REPEAT ko'rsatmasi

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- siklning *repeat* ko'rsatmasi takrorlanishlar sonini dastur bajarilishining borishi bilan aniqlab boradi;
- *repeat* sikl ko'rsatmasi *until* so'zidan keyingi shart yolg'on bo'lguncha bajariladi;
- *until* so'zidan keyin siklning tugash shartini yozish kerak;
- *repeat* siklini tugatish uchun sikl tanasida siklning tugashiga ta'sir qiluvchi ko'rsatma, albatta, bo'lishi va bajarilishi kerak;
- *repeat* siklida sikl tanasidagi ko'rsatmalar hech bo'lmaganda bir marta bajariladi;
- *repeat* sikli, odatda, taqribiy hisoblashlarni tashkil qilishda, izlash va klaviatura yoki fayldan kiritiladigan ma'lumotlarni qayta ishlashda ishlatiladi.



Izoh: masala shartlarida dasturchi kiritadigan sonlar yo'g'on shrift bilan yozilgan.



MASALALAR

97. Foydalanuvchi kiritgan butun son tub ekanligini tekshirish dasturini tuzing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Butun sonni kiriting va <Enter>ni bosing → 45

45 — tub son emas.

98. 1 dan 10 gacha bo'lgan sohadagi son «o'ylaydigan» va foydalanuvchiga 5 ta urinishda sonni topadigan dastur tuzing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

«Sonni top» o'yini.

Kompyuter 1 dan 10 gacha bo'lgan son «o'yladi».

Uni 5 urinishda toping.

Sonni kiriting va <Enter>ni bosing → 5

Yo'q

→ 3

Siz yutdingiz! Tabriklayman!

99. Berilgan vaqt oralig'idan keyin ovoz signali beruvchi dastur — taymer tuzing. Vaqt oralig'i miqdori dastur ishi vaqtida klaviaturadan kiritiladi.

100. Klaviaturadan kiritiladigan musbat sonlar o'rta arifmetigini topish dasturini yozing.

101. Klaviaturadan ketma-ket kiritilgan sonlardan maksimalini topish dasturini yozing.

102. Integralni trapetsiya usuli bilan taqribiy hisoblash dasturini yozing. Siklning har bir takrorlanishida dastur oraliq miqdorlari va kattaligini chiqarib borsin.



2.1.10. WHILE ko'rsatmasi

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- *while* so'zidan keyin yozilgan shart haqiqat bo'lguncha *while* siklining ko'rsatmalari bajariladi;
- *while* so'zidan keyin sikl ko'rsatmasining bajarilish shartini yozish kerak;
- *while* siklini tugatish uchun sikl tanasida sikl ko'rsatmasining bajarilish shartiga ta'sir etuvchi ko'rsatmalar bo'lishi shart;
- *while* sikli ba'zi hollarda bir marta ham bajarilmasligi mumkin;
- *while* sikli, odatda, taqribiy hisoblashlarni tashkil qilishda, izlash va klaviatura yoki fayldan kiritiladigan ma'lumotlarni qayta ishlashda ishlatiladi.



Izoh: masala shartlarida dasturchi kiritadigan sonlar yo'g'on shrift bilan yozilgan.



MASALALAR

103. Foydalanuvchi bergan aniqlik bilan π ni hisoblash dasturini yozing. Buning uchun quyidagidan foydalaning: $1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots$ qatorning xususiy yig'indisi yetarli darajada katta sondagi hadlarni qo'shganda $\pi/4$ ga yaqinlashib keladi. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan:

π hisoblash aniqligini bering $\rightarrow 0.001$

π sonining 0.001000 aniqlik bilan qiymati $\rightarrow 3.143589$

Qatorning 502 hadi qo'shilgan.

104. Ikkita butun sonning eng katta umumiy bo'luvchisini topish dasturini yozing.

105. a sonini b soniga bo'lishdagi qoldig'ini, ($a \leq 0$; $b > 0$) DIV va MOD amallaridan foydalanmay turib, aniqlang.

106. $e = a^4$ ni darajaga ko'tarish amalini bajarmasdan hisoblang.

107. 100 sonigacha bo'lgan juft sonlarning kublari va toq sonlarning kvadratlari yig'indisini hisoblang.

108. Berilgan tub sonni ko'paytuvchilarga ajrating.



2.1.11. Belgilar va qatorlar

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- har bir belgiga son — belgi kodi mos keladi;
- satrning alohida belgisiga yo'l topish uchun o'zgaruvchi (satrning) nomidan keyin kvadrat qavslarda uning raqamini ko'rsatish mumkin.



Izoh: masala shartlarida dasturchi kiritadigan sonlar yo'g'on shrift bilan yozilgan.



MASALALAR

109. Foydalanuvchidan uning nomini va sog'lig'ini so'rovchi dastur yozing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Ismingiz nima?

Ismingizni kiriting va `<Enter>`ni bosing

→ Eshmat

Assalomu alaykum, Eshmat!

110. «Telegraf» yo'sinida (ya'ni, axborot harflari bittadan, ba'zi bir sukut bilan paydo bo'lishi kerak) ekranga axborot chiqaruvchi dasturni yozing.

111. Foydalanuvchi tomonidan kiritilgan belgi kodini ASCII standartida chiqaruvchi dasturni yozing. Dastur ishini, masalan, nuqta kiritilgandan keyin tugatishi kerak. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Belgini kiriting va `<Enter>`ni bosing.

Dastur ishini tugatish uchun nuqtani kiriting → •

→ 1

Belgi: 1 kod: 81

→ 2

Belgi: 8 kod: 88

→ E

Belgi: E kod: 101

→ •

112. Belgilar kodi jadvalining birinchi qismini (0 dan 127 gacha kodga ega bo'lgan belgilar) ekranga chiqarish dasturini yozing. Jadval sakkizta ustun va o'n oltita satrdan iborat bo'lishi kerak. Birinchi ustunda 0 dan 15 gacha kodga ega bo'lgan belgilar, ikkinchisida 16 dan 31 gacha va h.k. bo'lishi kerak.

113. Klaviaturadan kiritilgan kirill imlosining yozma harflarini bosma harflarga aylantirish dasturini yozing (kirill imloli *UpCase* funksiyasi ishlamasligini e'tiborga oling). Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Matn satrini kiriting va <Enter>ni bosing.

→ Paskalni o'rgangach, Delfiga o'tish mumkin.

Yuqori registrda o'zgartirilgan satr:

PASKALNI O'RGANGACH, DELFIGA O'TISH MUMKIN.

114. Klaviaturadan kiritilgan satrning boshlang'ich bo'sh xonalarini olib tashlovchi dasturni yozing.

115. Klaviaturadan kiritilgan satr butun son ekanligini tekshiruvchi dasturni yozing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Sonni kiriting va <Enter>ni bosing.

→ 23.5

Kiritilgan satr butun son emas.

116. Klaviaturadan kiritilgan satr o'n oltilik son ekanligini tekshirish dasturini yozing.

117. Klaviaturadan kiritilgan satr kasr son ekanligini tekshirish dasturini yozing.

118. Klaviaturadan kiritilgan sakkiz xonali ikkilik sonni o'nlikka aylantirish dasturini yozing. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Sakkiz xonali ikkilik sonni kiriting va <Enter>ni bosing.

→ 11101010

Ikkilik 11101010 soniga o'nlik 324 soni mos keladi.

Dastur ishini tugatish uchun <Enter>ni bosing.

119. Klaviaturadan kiritilgan ikki xonali o'n oltilik sonni o'nlikka aylantirish dasturini yozing.

120. Foydalanuvchi tomonidan kiritilgan o'nlik sonni ko'rsatilgan sanoq tizimidagi (2 likdan 10 likkacha) songa o'tkazish dasturini yozing. Dastur ishlash vaqtida ekranning tavsiya qilinadigan ko'rinishi quyidagicha:

Butun sonni kiriting $\rightarrow 67$

Sanoq tizimi asosini kiriting $\rightarrow 2$

100011

121. Foydalanuvchi tomonidan kiritilgan o'nlik sonni o'n oltilikka o'tkazish dasturini yozing.

122. $N_0O_1, N_1O_2, \dots, O_kN_k$ ifoda qiymatini hisoblovchi dasturni yozing, bu yerda, N_i – butun bir xonali son, O_i – oddiy arifmetik amallarning (qo'shish yoki ayirish) ikki belgisidan biri. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Arifmetik ifodani kiriting, masalan, $4 + 5 - 3 - 5 + 2$ va $\langle \text{Enter} \rangle$ ni bosing.

$\rightarrow 9 - 5 + 4 + 2 - 6$

Kiritilgan ifoda qiymati: 4

Dastur ishini yakunlash uchun $\langle \text{Enter} \rangle$ ni bosing.



2.1.12. Massivlar

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- massiv bu bir xil turdagi elementlar to'plamini ifodalovchi ma'lumotlar tasnifidir;
- massiv o'zgaruvchilarni e'lon qilish bo'limida e'lon qilinishi kerak;
- massivlarni e'lon qilishda nomlangan o'zgaruvchilardan foydalanish qulay;
- massiv, elementiga yo'l element indeksi (raqami)ni ko'rsatish orqali ochiladi. Indeks sifatida butun turdagi ifoda, masalan, butun o'zgaruvchi yoki *integer* turidagi o'zgaruvchi ishlatiladi;
- massivlarni kiritish, chiqarish va qayta ishlashda sikl ko'rsatmalaridan (*for*, *while*, *repeat*) foydalanish qulay.



Izoh: masala shartlarida dasturchi kiritadigan sonlar yo'g'on shrift bilan yozilgan.



MASALALAR

123. Klaviaturadan beshta butun sondan iborat bir o'lchovli massivni kirituvchi va nolmas elementlar miqdorini hisoblovchi dasturni yozing. Har bir elementni kiritishdan oldin ekranda tartib raqami bilan belgi izohi chiqsin.

Har bir son kiritilgach, $\langle \text{Enter} \rangle$ bosilsin.

a [1] \rightarrow 12

a [2] \rightarrow 0

a [3] \rightarrow 3

a [4] \rightarrow -1

a [5] \rightarrow 0

Massivda nolmas elementlar uchta.

124. Klaviaturadan kiritilgan butun sonlar massividan eng kichik qiymatlisini topish dasturini tuzing. Quyida ish vaqtida tavsiya etiluvchi, ekran ko'rinishi berilgan.

Massivning minimal elementini topish.

Bir satrdan massiv elementlarini (5 ta butun son) kiriting va $\langle \text{Enter} \rangle$ ni bosing \rightarrow 23 0 45 -5 12

Massivning minimal elementi: -5

125. Kunning haftalik o'rtacha haroratini hisoblovchi dasturni yozing. Boshlang'ich qiymatlar dastur ishlash vaqtida kiritilishi mumkin. Quyida ish vaqtida tavsiya etiluvchi ekran ko'rinishi berilgan.

Haftalik havo harorati ma'lumotlarini kiriting.

Dushanba \rightarrow 12

Seshanba \rightarrow 10

Chorshanba \rightarrow 16

Payshanba \rightarrow 18

Juma \rightarrow 17

Shanba \rightarrow 16

Yakshanba \rightarrow 14

Haftalik o'rtacha harorat: 14.71 daraja.

126. Massivda klaviaturadan kiritilgan son bor-yo'qligini tekshiruvchi dasturni yozing. Ishlash vaqtida massiv kiritilishi kerak.

127. $a\{a_1, a_2, \dots, a_n\}$ haqiqiy qiymatli vektor berilgan. Uning hamma manfiy qiymatlarini kvadratlari bilan almashtiring.

128. $a\{a_1, a_2, \dots, a_n\}$ haqiqiy qiymatli vektor berilgan. Shu vektor elementlaridan (c, d) oraliqda yotgan elementlar yig'indisi va miqdorini toping.

129. $a\{a_1, a_2, \dots, a_n\}$ haqiqiy qiymatli vektor va X haqiqiy son berilgan. Vektor elementlarini tartibga soling va sonning elementlarning qaysi oralig'ida yotishini aniqlang.

130. $a\{a_1, a_2, \dots, a_n\}$ haqiqiy qiymatli vektor va X haqiqiy qiymatli son berilgan. Vektor elementlaridan nechitasi shu songa teng bo'lishini aniqlang.

131. $a\{a_1, a_2, \dots, a_n\}$ haqiqiy qiymatli vektor berilgan. Shu vektorning musbat elementlarini 0,5 ga kamaytirib boring.

132. Massivning klaviaturadan kiritilgan elementlarini o'sib borish ketma-ketligida berilganini tekshiruvchi dasturni yozing.

133. Klaviaturadan kiritilgan son massivda nechanchi o'rinda turishini aniqlovchi dasturni yozing.

134. $a\{a_1, a_2, \dots, a_n\}$ haqiqiy qiymatli vektor berilgan. Shu vektorda nechta musbat, nechta manfiy son va nechta nol borligini aniqlang.

135. $a\{a_1, a_2, \dots, a_n\}$ haqiqiy qiymatli vektor berilgan. Vektorning manfiy hadigacha bo'lgan elementlar miqdorini hisoblang.

136. $a\{a_1, a_2, \dots, a_n\}$ haqiqiy qiymatli vektor berilgan. «Y» o'zgaruvchiga a vektorning hamma musbat elementlari yig'indisining ikkilangan qiymatini o'zlashtiring.

137. $a\{a_1, a_2, \dots, a_n\}$ haqiqiy qiymatli vektor berilgan. Vektor toq raqamli elementlarini 5 ga, juft raqamli elementlarini 3 ga ko'paytirib, ularning yig'indisini toping.

138. $a\{a_1, a_2, \dots, a_n\}$ haqiqiy qiymatli vektor berilgan. Uning manfiy elementlarini (vektorda joylashish tartibini buzmasdan) birinchi, manfiy bo'lmagan elementlarini esa keyingi o'rinlarda joylashtirib, yangi $b\{b_1, b_2, \dots, b_n\}$ vektor hosil qiling.

139. Klaviaturadan kiritilgan bir o'lchovli massivni to'g'ri tanlov usuli bilan kamayish tarzida saralovchi dasturni yozing.

140. Klaviaturadan kiritilgan bir o'lchovli massivni almashtirish («ko'pik») usuli bilan kamayish tarzida saralovchi dasturni yozing.

141. O'sib borish tarzida tartibga solingan 2 massivni birlashtiruvchi dasturni yozing. Quyida ish vaqtida tavsiya etiluvchi ekran ko'rinishi berilgan.

Tartibga solingan ikkita massivni birlashtirish.

Bitta satrdan birinchi massiv elementlarini kiriting (5 ta butun son) → 1 3 5 7 9

Bitta satrdan ikkinchi massiv elementlarini kiriting (5 ta butun son) → 2 4 6 8 10

Natija — massiv

1 2 3 4 5 6 7 8 9 10

Ishni yakunlash uchun <Enter>ni bosing.

142. Binar izlash usulini qo'llab, o'sib borish tarzida tartibga solingan massivda izlashni amalga oshiruvchi dasturni yozing.

143. Guruhda bo'yi o'rtachadan baland bo'lgan talabalar sonini aniqlash dasturini tuzing. Quyida ish vaqtida tavsiya etiluvchi ekran ko'rinishi berilgan:

Talabalar bo'yi tahlili

Bo'yini kiriting (sm) va <Enter>ni bosing.

Ishni yakunlash uchun 0 kiriting va <Enter>ni bosing.

→ 175

→ 170

→ 180

→ 168

→ 170

→ 0

O'rtacha bo'y: 172.6 sm.

Ikki talabaning bo'yi o'rtachadan baland.

144. Klaviaturadan satrlar bo'yicha ikki o'lchovli massivni kirituvchi va uning ustunlar bo'yicha yig'indisini topuvchi dasturni yozing.

145. Ikkinchi tartibli kvadrat matritsa aniqlovchisini hisoblovchi dasturni yozing. Quyida ish vaqtida tavsiya etiluvchi ekran ko'rinishi berilgan:

Ikkinchi tartibli matritsani kiriting.

Satr elementlari kiritilgach, <Enter>ni bosing.

→ 5 - 7

→ 1 3

Matritsa aniqlovchisi:

5.00 - 7.00

1.00 3.00

22.00 ga teng.

146. Klaviaturadan kiritilgan kvadrat matritsa «afsunkor» (magik) kvadrat ekanligini tekshiring.

«Afsunkor» kvadrat deb, har bir satr, har bir ustun va har bir diagonal bo'yicha yig'indisi bir xil songa teng matritsaga aytiladi (quyiga qarang):

2	9	4
7	5	3
6	1	8

13	8	12	1
2	11	7	14
3	10	6	15
16	5	9	4

147. Universiada o'yinlari natijalarini yakunlovchi dastur yozing. Foydalanuvchi dasturga har bir ishtirokchi – guruh qo'lga kiritgan har xil darajadagi medallar sonini kiritishi, dastur esa medallarning

umumiy soni va ularga mos keluvchi ochkolarni hisoblashi, shundan keyin to'plangan ochkolar soniga mos ravishda (oltin medal uchun — 7, kumush uchun — 6, bronza uchun — 5 ochko oladi) ro'yxatni tartibga solishi kerak. Dasturning ishlash vaqtida ekranning tavsiya etiluvchi ko'rinishi quyida keltirilgan:

Universiada o'yinlari natijasi.

Bir satrda oltin, kumush va bronza medallari soni kiritilgan:

Andijon	→	3	6	9
Toshkent sh.	→	12	9	8
Jizzax	→	6	5	4
Navoiy	→	0	6	2
Namangan	→	3	1	2
Buxoro	→	10	10	5
Surxondaryo	→	9	6	3
Samarqand	→	6	3	4
Sirdaryo	→	2	4	6
Farg'ona	→	5	1	4
Xorazm	→	3	5	10
Qoraq. Resp.	→	3	4	5
Qashqadaryo	→	2	6	9
Toshkent vil.	→	2	2	1

Universiada o'yinlari yakunlari

T/r	Viloyatlar	Oltin	Kumush	Bronza	Jami	Ochkolar
1.	Toshkent sh.	12	9	8	29	178
2.	Buxoro	10	10	5	25	155
3.	Surxondaryo	9	6	3	18	114
4.	Andijon	3	6	9	18	102
5.	Xorazm	3	5	10	18	101
6.	Qashqadaryo	2	6	9	17	95
7.	Jizzax	6	5	4	15	92
8.	Samarqand	6	3	4	13	80
9.	Qoraq. Resp.	3	4	5	12	70
10.	Sirdaryo	2	4	6	12	68
11.	Farg'ona	5	1	4	10	61
12.	Navoiy	0	6	2	8	46
13.	Namangan	3	1	2	6	37
14.	Toshkent vil.	2	2	1	5	31

148. «Sonni top» o'yin dasturini tuzing. O'yin qoidalari quyidagicha. Ikki kishi o'ynaydi. Biri son o'ylaydi, ikkinchisi uni topadi. Har qadamda topuvchi-o'yinchi o'z farazini aytadi, sonni o'ylagan esa sonning nechta raqami topilganini va bu raqamlardan nechitasi sonda to'g'ri o'rinni egallaganini aytadi. Masalan, agar 725 soni o'ylangan va 523 taxmin kiritilgan bo'lsa, ikkita (5 va 2) soni topilgan va ulardan bittasi (2) to'g'ri o'rinni egallagan, deb hisoblanadi. Quyida ish vaqtida tavsiya etiluvchi ekran ko'rinishi berilgan.

Kompyuter uch xonali son o'yladi. Siz uni topishingiz kerak. Navbatdagi son kiritilgandan keyin Sizga nechta raqam topilgani va ulardan nechitasi o'z o'rinlarida ekanligi aytiladi.

Sonni kiritgandan keyin <Enter>ni bosing.

O'yinni tugatish uchun <Esc>ni bosing.

Sizning variantingiz → 123	Topildi: 0.	O'z o'rnida: 0.
Sizning variantingiz → 456	Topildi: 1.	O'z o'rnida: 0.
Sizning variantingiz → 654	Topildi: 2.	O'z o'rnida: 2.
Sizning variantingiz → 657	Topildi: 2.	O'z o'rnida: 2.
Sizning variantingiz → 658	Topildi: 3.	O'z o'rnida: 3.

*** Siz topdingiz ! ***

Ishni tugatish uchun <Enter>ni bosing.

149. Foydalanuvchidan axborot qabul qiluvchi va uni ekranga nuqta va tirelar ketma-ketligi ko'rinishida chiqaruvchi telegraf-dastur yozing. Nuqta va tirelarni, mos uzunlikda, ovoz signallari hamrohligida berish mumkin. Kirill harflarining Morze belgilari quyida keltirilgan:

А	.-	Б	...-	В	..---	Г	---.
Д	..--	Е	..	Ж	...-	З	---..
И	..	Й	..---	К	..-	Л	..--
М	---	Н	..	О	---	П	..---
Р	..-	С	...	Т	-	У	..--
Ф	..-	Х	Ц	..-	Ч	---
Ш	----	Щ	---.	Ъ	..-	Ы	---
Ъ	..-	Э	..-	Ю	..-	Я	..-



2.1.13. Funksiyalar

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- ma'lumotlarni funktsiyaga uzatishdan oldin faqat parametrlarni (global o'zgaruvchilar, ya'ni funktsiyadan tashqarida e'lon qilingan o'zgaruvchilarni ishlatish tavsiya etilmaydi) ishlatish kerak;

- funksiyaning bajariluvchi ko'rsatmalari ichida funksiya qiymatlarini o'zlashtirish ko'rsatmalari, albatta, bo'lishi kerak;
- funksiya chaqirish ko'rsatmasida har bir haqiqiy parametr turi (o'zgarmaslar yoki o'zgaruvchi) funksiyani e'lon qilishda ko'rsatilgan mos rasmiy parametr bilan bir xil bo'lishi kerak;
- agar funksiyani e'lon qilish ko'rsatmasida rasmiy parametr ismi oldida *var* so'zi bo'lmasa, funksiyani chaqirish vaqtida rasmiy parametr sifatida o'zgarmas yoki mos turdagi o'zgaruvchini ishlatish mumkin. Agar *var* so'zi ko'rsatilgan bo'lsa, rasmiy parametr sifatida faqat o'zgaruvchi bo'lishi mumkin.
- agar funksiya parametrlari natijani faqat funksiyani chaqirgan dasturga qaytarish uchun ishlatilsa, funksiyani e'lon qilishda tegishli parametr ismi oldida *var* so'zi bo'lishi kerak.



Izoh: masala shartlarida dasturchi kiritadigan sonlar yo'g'on shrift bilan yozilgan.



MASALALAR

150. Argument sifatida olingan ikkita butun sondan kattasini qaytaruvchi funksiyani yozing.

151. Ikkita butun sonni taqqoslovchi va taqqoslash natijasini $>$, $<$ yoki $=$ belgilardan biri ko'rinishida qaytaruvchi funksiyani yozing.

152. Ikkita rezistordan iborat zanjir qarshiligini hisoblovchi funksiyani yozing. Qarshilik miqdorlari va ulash turlari (ketma-ket yoki parallel) funksiya parametrlari bo'ladi. Funksiya parametrlarning to'g'riligini tekshirish kerak: agar ulash turi xato ko'rsatilgan bo'lsa, funksiya 1 ni qaytarishi kerak.

153. a^b qiymatni hisoblovchi funksiyani yozing, a va b sonlar ixtiyoriy kasr musbat sonlar bo'lishi mumkin.

154. Omonat bo'yicha daromadni hisoblovchi *Daromad* funksiyasini yozing. Omonat miqdori, yillik foiz ustama va omonat muddati (kunlar soni) funksiyaning boshlang'ich ma'lumotlari bo'ladi.

155. Agar argument sifatida funksiya tomonidan qabul qilingan belgi alfavitning o'nli harfi bo'lsa, *True* qiymatini qaytaruvchi *Unli* funksiyasini yozing.

156. Argument sifatida olingan satrdagi boshlang'ich bo'shliqlarni olib tashlovchi *LTrim* funksiyasini yozing.

157. Argument sifatida hosil qilingan o'zgartirilgan satrni yuqori registrga qaytaruvchi funksiyani yozing. E'tibor bering, agar bu funksiya *UpCase* (lotin imlosi uchun shunga o'xshash masala yechuvchi funksiya *Turbo Pascal*da shunday) deb atalsa, dasturchining funksiyasi *Turbo Pascal* funksiyasi o'rnini bosadi.

158. Kvadrat tenglamani yechuvchi funksiyani yozing. Tenglama koeffitsientlari va ildizlari funksiya parametrlari bo'lishi kerak. Funksiya qiymati uni chaqiruvchi dasturga tenglama ildizi borligi haqidagi axborotlarni uzatish uchun ishlatiladi: 2 — har xil ildizlar, 1 — bir xil ildizlar, 0 — tenglama yechimga ega emas. Bundan tashqari, funksiya boshlang'ich ma'lumotlarning to'g'riligini tekshirishi kerak. Agar boshlang'ich ma'lumotlar xato bo'lsa, funksiya 1 ni qaytaradi.

159. Klaviaturadan butun ikki xonali musbat sonni kiritishni ta'minlovchi funksiyani yozing. Foydalanuvchi qandaydir bir klavishni bosishi bilan unga mos keluvchi belgi ekranda, u faqat raqam bo'lgan holdagina, paydo bo'lishi kerak. Funksiya foydalanuvchiga kiritilgan sonni <BackSpace> klavishi yordamida tahrirlashga imkon berishi kerak. Agar foydalanuvchi hech bo'lmaganda bitta raqamni kiritgan bo'lsa, <Enter> klavishining bosilishi bilan funksiya ishini tugatish va foydalanuvchi kiritgan sonni uni chaqirgan protseduraga qaytarishi kerak.

160. Silindr hajmini hisoblovchi funksiyani yozing. Funksiya parametrlari silindr radiusi va balandligi.

161. Argument sifatida funksiya tomonidan qabul qilingan kirill imlosidagi harf undosh bo'lganda, *TRUE* qiymatini beruvchi *Undosh* funksiyasini yozing.

162. Darajaga ko'tarish funksiyasini yozing.

163. Sonning faktorialini hisoblovchi funksiya tuzing va uni

$$C_n^m = \frac{n!}{m!(n-m)!} \text{ ni hisoblashda ishlatting.}$$

164. Sessiya natijalari bo'yicha guruhlar o'rtacha balini topish dasturini tuzing. To'rt guruh bo'yicha baholar quyidagi matritsalarda keltirilgan: *A* (24,5), *B* (23,5), *C* (22,5), *D* (24,5).

165. *A* (10,12), *B* (15,10) va *C* (8,10) matritsalar berilgan. Ularning eng kichik elementlarini topish funksiyasini tuzing.

166. *A* (80), *B* (70), *C* (100) massivlarda keltirilgan tajriba natijalarining o'rta qiymatlarini funksiya tuzib, hisoblang.

167. *X* (5,6), *Y* (7,4), *Z* (5,7) berilgan massivlarning maksimal elementlarini topish funksiyasini tuzing. Natijalarni *P* (3) massivda keltiring.

168. *A* (40), *B* (30), *C* (60) butun sonli massiv elementlarining ichidan 3 ga bo'linadiganlarini funksiya tuzib, toping.

169. $ax^2 + bx - 4 = 0$; $y^2 - y + d = 0$; $2z^2 + dz - 1 = 0$
kvadrat tenglamalarning katta ildizlarini hisoblash funksiyasini tuzing va katta ildizlarni $B(3)$ massivga kiriting.



2.1.14. Protseduralar

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- protseduraga axborotni uzatish uchun undan tashqarida e'lon qilingan global o'zgaruvchilarni emas, balki parametrlarni ishlatish kerak;

- protsedurani chaqirish ko'rsatmasidagi har bir haqiqiy parametr (o'zgarmaslar yoki o'zgaruvchi) turi funksiyani e'lon qilishda ko'rsatilgan tegishli rasmiy parametrga mos kelishi kerak;

- agar protsedurani e'lon qilish ko'rsatmasida rasmiy parametr ismi oldida *var* so'zi bo'lmasa, protsedurani chaqirish vaqtida rasmiy parametr sifatida o'zgarmas yoki mos turdagi o'zgaruvchini ishlatish mumkin. Agar *var* so'zi ko'rsatilgan bo'lsa, rasmiy parametr faqat o'zgaruvchi bo'lishi mumkin.

- agar protsedurani chaqirgan protsedura argumenti natijani dasturga qaytarish uchun qo'llanilsa, argument nomi oldiga *var* so'zi yoziladi.



Izoh: masala shartlarida dasturchi kiritadigan sonlar yo'g'on shrift bilan yozilgan.



MASALALAR

170. Ekranga yulduzchalardan iborat satrni chiqaruvchi protsedurani yozing. Satr uzunligi (yulduzchalar soni) funksiya parametri bo'ladi.

171. Bir xil belgilardan iborat satrni chiqaruvchi protsedurani yozing. Satr uzunligi va belgi protsedura parametrlari bo'ladi.

172. Ekranga ramkani chiqaruvchi *Frame* protsedurasini yozing. Protsedura parametrlari sifatida ramaning yuqori chap burchak koordinatalari va uning o'lchami uzatilishi kerak.

173. Berilgan *a* va *b* sonlardan kattasini *X*, kichigini *Y* o'zlashtiradigan dastur tuzing. Masala *min* va *max* protseduralarini o'z ichiga olsin.

174. 1 dan *n* gacha bo'lgan sonlarning kublari kvadratlari yig'indisini hisoblovchi protsedura tuzing.

175. *Tartib* dasturiga ikkita o'zgaruvchilarning qiymatlarini o'zaro almashtiradigan protsedura tuzib kiriting.

176. Chiziqli tenglamalar tizim ildizlarini protseduradan foydalanib, Kramer formulasi yordamida toping.

177. $P_n(X) = a_n X^n + a_{n-1} X^{n-1} + \dots + a_1 X + a_0$ ko'phadni protseduradan foydalanib hisoblang.

178. $X = X_0, X_1, \dots, X_n$ (bu yerda, $X_{i+1} = X_i + h$ ($i = 1, 2, \dots, n$)) sonlarning kvadrat va kublarini hisoblashda protseduradan foydalaning.

179. $Z = X_i Y_j$ ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$) miqdorni hisoblash protsedurasini tuzing. Bu yerda X_i, Y_j mos ravishda X va Y butun sonli vektor elementlari. Z butun sonli matritsa elementlari.

180. $X = \{X_1, X_2, X_3\}$ vektor uzunligini topish uchun protseduradan foydalaning. Uning hadlari mos ravishda $A(4, 4)$, $B(3, 3)$, $C(2, 2)$ matritsalar hadlarining yig'indisiga teng.



2.1.15. Fayllar

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- natijalarni faylga chiqaruvchi yoki boshlang'ich ma'lumotlarni fayldan o'quvchi dasturda text turidagi faylli o'zgaruvchi e'lon qilingan bo'lishi kerak;

- aniq faylga erishish uchun faylli o'zgaruvchini bu fayl bilan bog'lash kerak (bu *assign* ko'rsatmasi yordamida bajariladi);

- fayldan foydalanish mumkin bo'lishi uchun u ochiq bo'lishi kerak (*reset* — o'qish uchun, *rewrite* — yozish uchun, *append* — qo'shimcha kiritish uchun);

- fayllar bilan ishlashda xatoliklar bo'lishi mumkin, masalan, dastur, bo'lmagan faylni ochishga uringan paytda. Shuning uchun xatoliklarga olib kelishi mumkin bo'lgan har bir ko'rsatmadan keyin, *IOResult* funksiyasini qo'llab, fayl bilan ishni tugatish kodini tekshirish kerak: dastur fayl bilan bajariladigan ish natijasini nazorat qila olishi uchun, uning matniga *{SI —}* direktivani kiritish kerak;

- faylga yozishni *write* va *writeln* ko'rsatmalar, o'qishni *read* va *readln* ko'rsatmalar bajaradi, bunda bu ko'rsatmalarning birinchi parametri sifatida fayl o'zgaruvchisini ko'rsatish kerak;

- fayl bilan ish tugagandan keyin uni, albatta, *close* ko'rsatmasi bilan yopish kerak;

- dasturda o'zgaruvchisining turi *text* deb e'lon qilingan va shu dastur tomonidan yaratilgan faylni matn muharriri yordamida ko'rib chiqish mumkin.



Izoh: masala shartlarida dasturchi kiritadigan sonlar yo'g'on shrift bilan yozilgan.

181. Kompyuterning almashtirish diskida (A:) *numbers.txt* faylini hosil qiluvchi va unga foydalanuvchi tomonidan kiritilgan 5 ta butun sonni yozuvchi dastur tuzing. *Norton Commanderga* kiritilgan matn muharriri yordamida yaratilgan faylni ko'ring, har bir son alohida satrda joylashganligiga ishonch hosil qiling.

182. *a:numbers.txt* fayliga foydalanuvchi tomonidan kiritilgan 5 ta butun sonni yozuvchi dasturni yozing. Matn muharriri yordamida faylda 10 ta son yotganligini tekshiring.

183. *a:numbers.txt* fayl tarkibini ekranga chiqaruvchi dasturni yozing.

184. *a:numbers.txt* faylida joylashgan o'rta arifmetik qiymatni hisoblovchi dasturni yozing.

185. Matn fayllarini, masalan, *Turbo Pascal* boshlang'ich dasturlar fayllarini, ko'zdan kechirishga (ekranga fayl tarkibini chiqaradi) imkon beradigan dastur yozing. Ko'zdan kechiriladigan fayl ismi dastur ishlash vaqtida kiritilishi kerak. Tavsiya qilinadigan ekran ko'rinishi quyida keltirilgan:

Matn faylini ko'zdan kechirish.

Faylning to'la ismini kiriting va <Enter> klavishini bosing →
`c:\tp\ pas \numb.pas`

186. A: diskda joylashgan *phone.txt* fayliga o'rtoq'ingiz ismi-sharifi va telefon raqamini yozuvchi dasturni yozing. Agar fayl diskda bo'lmasa, dastur uni yaratishi kerak. Faylda ma'lumotlarning har bir elementi (ismi-sharifi, telefon raqam) alohida satrda bo'lishi kerak. Dasturning ishlash vaqtida tavsiya etiladigan ekran ko'rinishi quyida keltirilgan.

Telefon ma'lumotnomasiga qo'shimcha

Familiyasi →

Ismi →

Telefon →

Axborot qo'shildi.

Ishni tugatish uchun <Enter>ni bosing.

187. Telefon ma'lumotnomasida (*a: phone.txt*) ma'lumotlarni topishga imkon beruvchi dasturni yozing. Dastur kishi familiyasini so'rashi va uning telefon raqamini chiqarishi kerak. Agar ma'lumotnomada bir xil familiyalar bo'lsa, dastur shunday familiyaga ega hamma kishilarning ro'yxatini chiqarishi kerak. Dasturning ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi quyida keltirilgan.

Telefon ma'lumotnomasidan izlash.

Familiyani kiritish va <Enter>ni bosish. Dastur bilan ishni tugatish uchun taklifdan keyin darrov <Enter>ni bosish.

Familiyasi → Yusupov.

Ma'lumotnomada Yusupov haqida axborot yo'q.

→ Isayev.

Isayev Hamro 578 – 12 – 45

Isayev Hasan 224 – 34 – 02

188. Testdan o'tkazish universal dasturini yozing. Test savollari ketma-ketligi va javob variantlari matn faylida bo'lishi kerak. Dastur bu fayl ismini uni ishga tushiruvchi buyruq satridan olishi kerak. Test savollari soni cheklanmagan. Lekin quyidagi cheklilarni kiritish taklif etiladi: savol matni va muqobil javoblar ekranning bitta satridan ortiq joyni egallamasligi kerak.

Dastur baholarni quyidagi qoidaga ko'ra qo'yish kerak. Hamma javoblar to'g'ri bo'lsa — a'lo, 80 % dan kam bo'lmagan savollarga to'g'ri javob berilsa — yaxshi, 60 % dan kam bo'lmagan to'g'ri javoblarga qoniqarli, 60 % dan kam to'g'ri javoblar uchun yomon.

Quyida test savollari faylining tavsiya etiladigan tasnifi keltiriladi (N_i — i — savolga muqobil javoblar soni; K_i — to'g'ri javob raqami), test fayliga misol va ekranning dastur ishlayotgan vaqtidagi ko'rinishi quyida keltirilgan.

Savol₁

$N_1 M_1$

Javob

...

Javob

Savol₂

$N_2 M_2$

Javob

...

Javob

Savol_k

$N_k M_k$

Javob

...

Javob

div nimani bildiradi?

3.2.

bo'lishni

butun bo'linmani
bo'lishdan hosil bo'lgan qoldiqni
Paskal tili asoschisi kim?

2.2.

Norbert Viner

Niklas Virt

Quyidagi mantiqiy amallardan qaysi biri birinchi bo'lib
bajariladi:

3.2

and

not

or

Hozir Sizga test taklif etiladi. Har bir savolga bir nechta javoblar
varianti beriladi. Siz to'g'ri javob raqamini kiritishingiz va <Enter>ni
bosishingiz kerak.

div nimani bildiradi?

1. Bo'lishni.
2. Butun bo'linmani.
3. Bo'lishda hosil bo'lgan qoldiqni.

→ 2

Paskal tili asoschisi kim?

1. Norbert Viner.
2. Niklas Virt.

→ 2

Quyidagi mantiqiy amallardan qaysi biri birinchi bo'lib
bajariladi?

1. *and*.
2. *not*.
3. *or*.

→ 2

Bahongiz: «A'lo!».

Dastur ishini yakunlash uchun <Enter>ni bosing.

189. Duymlardan millimetrlarga o'tish jadvalini, foydalanuvchi
xohishiga ko'ra, ekranga, printerga yoki faylga chiqarish dasturini
yozing. Quyida dasturning ishlash vaqtida tavsiya qilinadigan ekran
ko'rinishi keltirilgan.

*** Duymlardan millimetrlarga o'tish jadvali ***

Natija chiqariladi:

1 – ekranga

2 – printerga

3 – faylga

1 dan 3 gacha sonlarni kiriting va <Enter>ni bosing.

Sizning tanlovingiz → 2

Duymlar	Millimetrlar
0.5	12.7
1.0	25.4
1.5	38.1
2.0	50.8
2.5	63.5

Duymlar	Millimetrlar
3.0	76.2
3.5	88.9
4.0	101.6
4.5	114.3
5.0	127.0

190. a: \phone.txt fayliga bir seansda bir nechta odamlar to'g'risida axborot kiritish dasturini yozing.

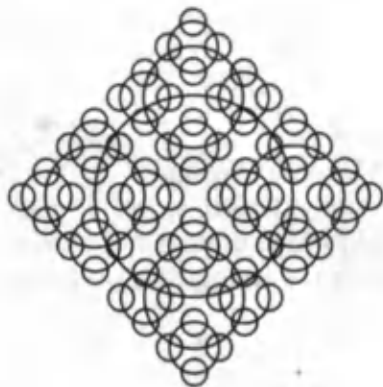
191. Telefon ma'lumotnomasiga qo'shimcha kiritadigan, ma'lumotnomada izlash ishlarini birlashtiradigan dastur tuzing.



2.1.16. Rekursiya

Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

- o'z-o'zini chaqirib oluvchi protsedura va funksiya rekursiv deyiladi;
- jarayonni tugatish uchun rekursiv funksiya (protsedura) algoritmidagi funksiyaning (protseduraning) bevosita tugashini ta'minlovchi yo'nalish, albatta, bo'lishi kerak.

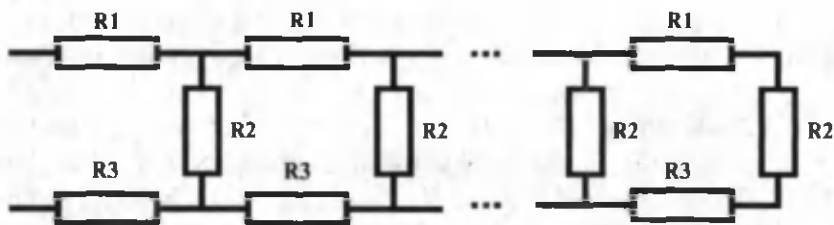


1-rasm.

192. Faktorialni hisoblovchi funksiyani va uning ishlashini tekshiruvchi dasturni yozing.

193. 1-rasmda keltirilgan bezakni ekranga chiqaruvchi dasturni yozing.

194. Chizmasi keltirilgan elektr zanjir qarshiligini hisoblash dasturini yozing. Qarshiliklar miqdori va zanjir tartibi (R_2 qarshiliklar soni) dastur ishlashi vaqtida kiritilishi kerak (1-chizma).



1-chizma.

195. Yuqorida keltirilgan elektr zanjir chizmasini ekranga chizuvchi dasturini yozing. Zanjir tartibi dastur ishlash vaqtida kiritilishi kerak.

196. Yig'indini X ning haqiqiy qiymatlarida hisoblang:

$$S = \sum_{n=1}^k \frac{X^{2n}}{(2n)}.$$

197. Yig'indini X ning haqiqiy qiymatlarida hisoblang:

$$S = \sum_{n=1}^k \frac{X^{4n+1}}{4n+1}.$$

198. Yig'indini X ning haqiqiy qiymatlarida hisoblang:

$$S = \sum_{n=1}^k (-1)^n \frac{\cos nx}{n^2}.$$

199. Yig'indini X ning haqiqiy qiymatlarida hisoblang:

$$S = \sum_{n=1}^k \frac{n^2 + 1}{n!} \left(\frac{X}{2}\right)^n.$$

200. $y=e^x$ funksiyaning qiymatini argumentning quyidagi o'zgarishlar sohasining $1 \leq X \leq 2$ har bir nuqtasida qatorga yoyib ($n=15$) hisoblang. Qator yig'indisining umumiy hadi:

$$S = \sum_{n=1}^k \frac{X^n}{n!}.$$

2.2. YUQORI MURAKKABLIKDAGI MASALALAR

Bu masalalarni berishdan maqsad qobiliyatli talabalarni dasturlash bilan shug'ullanishga qiziqtirish, ularga o'z ustlarida ishlashlari uchun ma'lumot berish — ularning ko'nikma, mahorat, qobiliyatlari, kasb malakasini oshirish.

201. «Ko'ndalang diagonal». $A(m, n)$ massivning, indekslari ayirmasi berilgan K soniga (K — manfiy son ham bo'lishi mumkin) teng bo'lgan ($i-j=k$), elementlari yig'indisini toping.

202. «Kvadratchalar». Har bir elementi 0,1,5 yoki 11 ga teng $A(m, m)$ massiv berilgan. Har birida elementlari har xil bo'lgan to'rtliklar $A(i, j)$, $A(i+1, j)$, $A(i, j+1)$, $A(i+1, j+1)$ miqdorini toping.

203. «Sanoq sistemalari». $M(9)$ massivda qandaydir natural sonning l — sanoq tizimida raqamlari yozilgan. ($M(l)$ — birlar xonasi va h.k.). Birlar xonasidan boshlab turib, J — sanoq tizimida bu son raqamlarini chop eting. I, J sonlar 10 dan oshmaydi.

204. «Kalendar». Kun, oy, yilni bildiruvchi uchta a, b, c sonlar berilgan. Shu kunning yil boshidan hisoblangandagi n -tartib raqamini toping.



Ko'rsatma: tartib raqami 400 ga bo'linadigan, shuningdek, tartib raqami 4 ga bo'linib, 100 ga bo'linmaydigan yillar kabisa yili hisoblanadi.

205. «So'zning matnga kiritilishi». Butun sonli $X(n)$ va $Y(k)$ 2 ta massivlar berilgan. $X_{i+1} = Y_i$, $X_{i+2} = Y_2$, ..., $X_{i+k} = Y_k$ shartlarni qanoatlantiruvchi va X_{i+j} , X_{i+2j} , ..., X_{i+kj} ketma-ket keluvchi « k » ta elementlarni birinchi massivdan tanlab olish mumkinmi? Bu masalani yechuvchi va «Ha» yoki «Yo'q» javoblarni beruvchi dasturni yozing.

206. «Nollar seriyasi». $A(n)$ butun qiymatli massiv berilgan. Massivning elementlari nolga teng va ketma-ket keluvchi eng uzun ketma-ketligi uzunligini toping.

207. «Egar nuqta». $A(m, n)$ sonli massiv berilgan. O'z satrida eng kichik bo'lgan element o'z ustunida eng katta bo'lsa, u egar nuqta deyiladi. Agar massivda egar nuqta bo'lsa, u yotgan satr va ustun raqamini, agar unday nuqta bo'lmasa, nolni chop eting.

208. «Kasrni qisqartirish». Natural « m » va « n » sonlar berilgan. Umumiy bo'luvchiga ega bo'lmagan shunday natural $m1$ va $n1$ sonlarni topish kerakki, $m1/n1 = m/n$ bo'lsin.

209. «Massivlarning qo'shilishi». « M » va « N » sonlar va ikkita tartibga solingan massivlar: $a_1 \leq a_2 \leq \dots \leq a_m$ hamda $b_1 \leq b_2 \leq \dots \leq b_n$ berilgan.

Bu elementlardan tartibga solingan uchinchi massivni hosil qiling: $c_1 \leq c_2 \leq \dots \leq c_{m+n}$. Ko'rsatma: « M » va « N »lar katta sonlar bo'lganda dasturdagi amallar miqdoriga e'tibor bering.

210. «Moda». Butun sonli $A(n)$ massivda eng ko'p uchraydigan sonni toping. Agar bunday sonlar bir nechta bo'lsa, ulardan bittasini aniqlang.

211. «Markaziy qishloq». « K » ta qishloq mavjud. Agar i — qishloqqa tez yordam punkti joylashtirilsa, chaqiruv bo'yicha j — qishloqqa borish $a_{ij} + a_{ji}$ ($1 \leq i, j \leq k$, $i \neq j$) vaqtini oladi. Shunday i — qishloq tartib raqamini topingki, undan eng uzoq qishloqqa borish uchun kam vaqt sarflansin. $A(k, k)$ massiv berilgan. Unda hamma a_{ii} elementlar noldan katta va a_{ii} — element a_{jj} elementga teng bo'lmashligi mumkin.

212. «Tartib indeksleri». $A(n)$ sonli massiv berilgan. $1, 2, \dots, n$ sonlarning shunday, i_1, i_2, \dots, i_n o'rin almashtirishini topingki, natijada $a_{i_1} \leq a_{i_2} \leq \dots \leq a_{i_n}$ bo'lsin.

213. «Nollashtirish». Berilgan ikki o'lchovli $A(m, n)$ massivda noli bo'lgan satr va ustun elementlarini nollar bilan almashtiring.



Shart: yordamchi bir o'lchovli massivdan foydalanish mumkin, lekin yordamchi ikki o'lchovli massiv ishlatish mumkin emas.

214. «Ulgurji xarid». Paypoq jufti 105 so'm, bog'lami (12 juft) 1025 so'm, qutisi (12 bog'lam) 11400 so'm turadi.

Xaridor sotib olmoqchi bo'lgan paypoqlarning n juft soniga ko'ra, xaridor sotib olishi kerak bo'lgan $n1, n2, n3$ quti, bog'lam, paypoqlar juftini hisoblab beruvchi dastur tuzing.



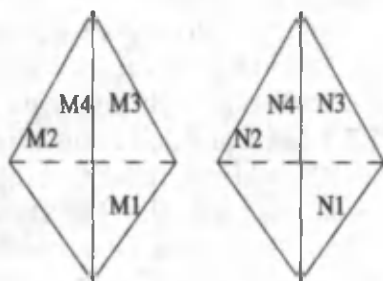
Tushuntirish. 11 juft paypoq o'rniga bir bog'lamni xarid qilish arzonga tushadi.

215. «To'ntarilgan sonlar». $A(N)$ sonli massiv berilgan. Massivning maksimal uzunlikdagi kesmasini topish kerak. Unda birinchi son oxirgisiga, ikkinchi son oxirgisidan bitta oldingisiga va h.k. teng bo'lsin. Bu kesma uzunligini chop eting.

216. «Ikki marta monoton». Sonlarning $A(m, n)$ massivi satrlar va ustunlar bo'yicha kamayib borish tarzida tartiblangan, ya'ni hamma $i=1, \dots, m$ lar uchun, $a_{i1} \leq a_{i2} \leq \dots$ hamma $j=1, \dots, n$ lar uchun $a_{j1} \leq a_{j2} \leq \dots$. Massiv elementlari ichidan berilgan «x» soniga teng bo'lganini toping.

Agar bunday element bo'lmasa, «Yo'q» deb chop eting. Majburiy shart. Yechimda amallar soni $m * n$ atrofida emas, $m+n$ atrofida bo'lsin.

217. «Tetraedrlar». Ikki ta teng to'g'ri M va N tetraedrlar qirrasiga $M1, M2, M3, M4$ va $N1, N2, N3, N4$ sonlar 2-chizmada ko'rsatilgan tartibda yozilgan.



2-chizma.

Tetraedrlarni bir xil sonlar yozilgan qirralari bilan bir-biriga mos tushirib, joylashtirish mumkinmi? «Ha» yoki «Yo'q» javob berilsin.

218. «Kasr davri». « M » va « N » natural sonlar berilgan. O'nlik M/N kasr davrini chop eting. Masalan, $1/7$ kasr uchun davr 142857 ga teng, kasr chekli bo'lsa, uning davri bitta 0 raqamiga teng bo'ladi.

219. «Arra». $X(m)$ massiv berilgan. Eng uzun «arra» shaklidagi (tishlari yuqoriga qaragan) ketma-ket keluvchi sonlar qatori uzunligini toping.

$$X[p+1] < X[p+2] > X[p+3] < \dots > X[p+k].$$

220. «Nollar oxiriga». Bir o'lichovli massiv berilgan. Uning nolga teng bo'lmagan hamma elementlarini, tartibni saqlagan holda, massiv boshiga, nollik elementlarini, yangi massiv ishlatmasdan turib, massiv oxiriga joylashtiring.

221. «Har xil raqamlik sonlar». O'nlik sanoq tizimidagi sonlar ichidan ikkita raqami bir xil bo'lmagan barcha to'rt xonali natural sonlarni chop eting.

222. «0,1,2 larning o'rnini almashtirish». $X(n)$ massivda har bir element 0,1 yoki 2 ga teng. Massiv elementlari o'rnini shunday almashtiringki, natijada avval hamma nollar, keyin birliklar, nihoyat, hamma ikkiliklar joylashsin (qo'shimcha massiv ishlatish mumkin emas).

223. «Arifmetik amallar». Yozilgan (((((1?2)?3)?4)?5)?6 ifodada har bir «?» belgi o'rniga +, -, *, /, arifmetik amallardan bittasini shunday qo'yish kerakki, hisoblash natijasi 35 bo'lsin (bo'lishda bo'linmadagi kasr qism tashlanadi). Bir yechimni topish yetarli.

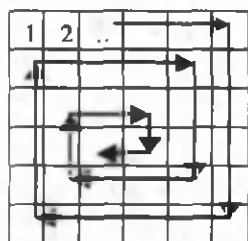
224. «Tez daraja». «A» haqiqiy va «K» natural sonlar berilgan. Darajaga ko'tarish amalidan foydalanmasdan (bu yerda, «K» juda katta son bo'lishi mumkin, natijada «K» ko'paytirishni bajarib bo'lmay qoladi) « A^k »ni hisoblang va chop eting.

225. «Har xil sonlar». « M » natural son va $A(m)$ butun qiymatli massiv berilgan. Bu massivda nechta har xil sonlar bor? Masalan, 5,7,5 dan iborat uch elementli massivda har xil sonlar ikkita (5 va 7).

226. «Minimumlarning maksimumi». $X(m,n)$ massivda hamma sonlar har xil. Har bir satrdan eng kichik elementni tanlash, so'ngra bu kichik elementlarning ichidan eng kattasini aniqlash dasturini tuzing. X — massivda tanlangan son yotgan satr va ustun raqamlarini aniqlang.

227. «Spiral». N sonini kiriting va $n \times n$ o'lchovli ikki o'lchamli massivni spiral bo'yicha $1, 2, \dots$ natural sonlari bilan to'ldiring (3-chizma).

228. «Ichki to'plam bo'yicha yig'indi». Butun qiymatli $A(n)$ massiv va « M » son berilgan. Elementlarning shunday qism to'plamini $A[i_1], A[i_2], \dots, A[i_k]$ ($1 \leq i_1 < \dots < i_k \leq n$) topish kerakki, $A[i_1] + A[i_2] + \dots + A[i_k] = M$ bo'lsin. Bunday to'plam bor, deb faraz qilinadi.



3-chizma.

229. «Labirint». Yo'lovchi labirintdan (boshi berk yo'l) chiqa oladimi? Agar chiqa olsa, unda yo'lovchining chiqishdan boshlang'ich nuqtasigacha bo'lgan yo'lini chop eting. Labirint 40×40 o'lchamli « A » massiv bilan berilgan. Unda agar (k, m) katakdan o'tish mumkin bo'lsa, $a[k, m] = 0$, (k, m) katakdan o'tish mumkin bo'lmasa, $a[k, m] = 1$. Yo'lovchining boshlang'ich nuqtasi, o'tish mumkin bo'lgan $[i, j]$ katakda. Yo'lovchi, agar kataklar bitta umumiy tomonga ega bo'lsa, bir katakdan ikkinchisiga o'ta oladi. Yo'lovchi chegara katakka (ya'ni (k, m) , bu yerda « k » va « m » 1 yoki 40 ga teng) yetsagina labirintdan chiqadi.

230. «Mukammal sonlar». Agar natural son o'zining barcha bo'luvchilari (1 ham hisoblanadi) yig'indisiga teng bo'lsa, u mukammal son bo'ladi. Berilgan M sonidan kichik barcha mukammal sonlarni topish dasturini tuzish.

231. «Hosil qilinmaydigan son». $P(n)$ natural sonlar massivi berilgan. Shunday eng kichik natural sonni topish kerakki, uni P massivning hech bir elementlari yig'indisi ko'rinishida ifodalab bo'lmasin. Yig'indi bitta qo'shiluvchidan ham iborat bo'lishi mumkin, lekin massivning har bir elementi unga faqat bir marta kira oladi.

232. «Do'mbira». Aylana bo'yicha 12 ta son yozilgan: a_1, a_2, \dots, a_{12} . Agar k — tartib raqamidan boshlab, ularni qo'shib yozsak, X_k vektor hosil bo'ladi:

$$X_k = (a_k, a_{k+p}, a_{k+1+p}),$$

Bu yerda, a_{13} ni a_1 , a_{14} ni a_2 va h.k. bildiradi. Agar birinchi tengmas juftdayoq $a_{k+j} < a_{p+i}$ ($i=0, 1, \dots$) bo'lsa, X_k vektor X_p vektordan kichik deb hisoblanadi. Shunday « k » ni topish kerakki, vektor X_k eng kichik bo'lsin.

233. «Kublar yig'indisi». Berilgan N natural sonni 2 ta natural sonlar kublarining yig'indisi ($N=i^3+j^3$) ko'rinishida necha xil usul bilan yozish mumkin?

Qo'shiluvchilarning o'rin almashtirilishi yangi usulni bermaydi. $1/3$ -darajaga ko'tarish amalidan foydalanish mumkin emas.

234. «Tub bo'luvchilar». Natural N son berilgan. Uning barcha tub bo'luvchilarini toping.

235. «Ko'phad». Berilgan haqiqiy X_1, X_2, \dots, X_n ildizlarga ega $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + x^n$ ko'phad berilgan. Uning a_0, a_1, \dots, a_{n-1} koeffitsientlarini toping.

! *Eslatma: Bezuz tesemasiga ko'ra $P(x) = (x-x_1)^{\alpha_1}(x-x_2)^{\alpha_2} \dots (x-x_n)^{\alpha_n}$.*

236. «Qisqa ko'paytuvchilar» A sonlar to'plami quyidagi shartlar bilan berilgan:

a) $1 \in A$;

b) agar $k \in A$ bo'lsa, $2^*k + 1 \in A$ va $3^*k + 1 \in A$ bo'ladi va A to'plam boshqa sonlarga ega bo'lmaydi. A to'plamning dastlabki $n < 1000$ sonlarini o'sib borish tartibida chop eting. Chop etiladigan sonlarning boshlanishi quyidagicha: 1, 3, 4, 7, 9, 10, 13, 15, 19, ...

237. «Poker». Beshta sondan iborat massiv berilgan. Ularning ichidan beshtasi bir xil bo'lsa — 1; to'rttasi bir xil bo'lsa — 2; uchta va ikkitasi bir xil bo'lsa — 3; uchta bir xil bo'lsa — 4; ikkita ikkitalik bir xil bo'lsa — 5; ikkitasi bir xil bo'lsa — 2; aks holda 7 sonini chop eting.

238. «Teng elementlar». Butun sonli $A(m, n)$ massiv berilgan. Massivning har bir satri kamayib borish tarzida tartibga solingan, ya'ni: hamma i lar uchun $i=1, \dots, m$; $a_{i1} \leq a_{i2} \leq \dots$ hamma satrlarda uchraydigan sonni toping va chop eting. Agar bunday son bo'lmasa, «yo'q» javobini hosil qiling.

239. «Yo'l». m va $n \geq 2$ natural sonlar hamda haqiqiy $A(m, m, n-1)$ massiv berilgan. Butun sonlarning mumkin bo'lgan hamma $1 \leq i_1, i_2, \dots, i_n \leq m$ to'plamlari uchun quyidagi yig'indining minimal qiymatini toping: $R = A(i_1, i_2, 1) + A(i_2, i_3, 2) + A(i_{n-1}, i_n, n-1)$.

! *Izoh: m va n sonlar bir necha o'nalikda teng miqdordir. Shuning uchun m^n tartibli amallar sonigacha yechish noo'rin.*

240. «Uchburchak va nuqta». Uchburchak uchlarining to'g'ri burchak koordinatalari $X_1, Y_1; X_2, Y_2; X_3, Y_3$ va nuqtaning X, Y koordinatalari berilgan. Nuqta uchburchakda yotish-yotmasligini aniqlang. Hisoblash xatoliklarini e'tibordan chetda qoldiring.

241. «Tartiblangan kasrlar». Maxraji 7 dan katta bo'lmagan qiymati 0 va 1 o'rtasida bo'lgan qisqarmaydigan hamma oddiy kasrlarni o'sib borish tartibida chop eting.

242. «To'rtliklar jufti». Yagona hol bo'lmagan to'rtta natural sonlar kvadratlari yig'indisini ifodalovchi eng kichik sonni toping.

243. « $M+1$ sonni ikkilikda yozish». Butun manfiymas « M » son o'zining ikkilik a_0, a_1, \dots, a_{n-1} raqamlari bilan berilgan $M = a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + \dots + a_1 \cdot 2 + a_0$ bu yerda a_i nolga yoki birga teng ($i=0, 1, \dots, n-1$). $M+1$ sonining ikkilik raqamlar massivini yozing.

244. «Kvadratlar yig'indisi». Berilgan M natural sonni ikkita natural sonlar kvadratlari yig'indisi ko'rinishida yozish mumkinmi?

245. «O'rin almashtirish». Har xil sonlardan iborat « M » ta juft sonlarning $A(M)$ massivi berilgan. Sonlarning barcha o'rin almash-tirishlarini hosil qiling.

246. «Tub sonlar». Berilgan M sonidan katta bo'lmagan barcha tub sonlarni bosmadan chiqaring.

247. «Teng sonlarni izlash». Butun qiymatli 2 o'lchamli $A(2, 15)$ massiv berilgan. Uning elementlaridan ikkitasi va faqat ikkitasining o'zaro tengligi ma'lum. Ularning indeksini aniqlang.

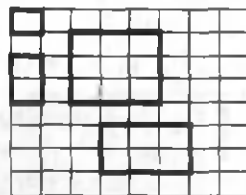
248. «Raqamlarning berilgan yig'indisi». Raqamlar yig'indisi berilgan natural songa teng bo'lgan barcha uch xonalik o'nlik sanoq tizimidagi sonlarni aniqlang.

249. «Funksiya». Butun manfiymas « N »lar uchun funksiya quyidagicha aniqlangan: $f(0)=0, f(1)=1, f(2n)=f(n), f(2n+1)=f(n)+f(n+1)$. Berilgan N uchun $f(N)$ ni toping. Zaruriy shart: N shunchalik katta sonki, N ta sondan massiv hosil qilish mumkin emas (M sonning o'sishi bilan massiv ham o'saveradi).

250. «To'rtburchaklar». 100×100 katakchali o'lchamdagi qog'oz varaqqa bir nechta to'rtburchaklar chizilgan. Har bir to'rtburchak butun kataklardan iborat, har xil to'rtburchaklar bir-birining ustiga yotmaydi va bir-biriga urilmaydi (4-chizma). 100×100 o'lchovli massiv berilgan, unda agar $[i, j]$ katak qandaydir bir to'rtburchakka tegishli bo'lsa, $A[i, j]=1$ va aks holda $A[i, j]=0$. To'rtburchaklar sonini hisoblab, chop etuvchi dasturni yozing.

251. «Bit — revers». Butun musbat « M » soni ikkilik sanoq tizimida yoziladi va bu yozuvda xonalar teskari tartibda, o'nlik sanoq tizimida qo'yiladi. Hosil bo'lgan son, $B(M)$ funksiya qiymati, deb qabul qilinadi. $M=512, 513, 514, \dots, 1023$ uchun $B(M)$ funksiya qiymatini chop eting. Aniqlik uchun chop etiladigan qiymatlar boshi: 1, 513, 257, ...

252. «Inversiya». $1, 2, \dots, n$ sonlarining o'rin almashtirishi $P=(P_1, \dots, P_n)$ bo'lsin. P o'rin almashtirishning inversiya jadvali shunday



4-chizma.

$T=(t_1, \dots, t_n)$ ketma-ketlikka teng bo'ladiki, unda « t_i », P o'rin almashtirishning i sonidan chapdagi « i » dan katta bo'lgan elementlar soniga teng bo'ladi. Masalan: 1, 2, ..., 9 sonlarining o'rin almashtirishi $P=(5, 9, 1, 8, 2, 6, 4, 7, 3)$ bo'lsin, uning inversiya jadvali quyidagicha hosil qilinadi: $i=1$ chapda bu sondan katta 2 ta son (5, 9) bor, demak, $t_1=2$; $i=2$, chapda bu sondan katta 3 ta son (5, 9, 8) bor, demak, $t_2=3$ va shunday davom ettiriladi. Natijada inversiya jadvali quyidagiga teng bo'ladi: $T=(2, 3, 6, 4, 0, 2, 2, 1, 0)$. Berilgan inversiya jadvaliga ko'ra sonlar o'rin almashtirishini tiklash dasturini tuzing.

253. «Qo'shiluvchilarga ajratish». Berilgan N natural sonni qo'shiluvchilarga ajratilgan hamma ko'rinishlarini hosil qiling. Qo'shiluvchilarning o'rin almashtirilishi yangi ko'rinish deb hisoblanmaydi.

254. «O'ng katta». Musbat sonlarning $A(N)$ massivi berilgan. Har bir a_i uchun a_j elementni tanlaymiz, a_j element quyidagicha tanlanadi: u a_i dan keyin kelishi, a_j dan katta va j -tartib raqami bo'yicha eng kichik bo'lishi kerak. a_j qiymat a_i elementi bilan almashtiriladi. Agar shunday a_j element bo'lmasa, a_j qiymat nol bilan almashtiriladi. Hosil bo'lgan massiv chop etilsin.

! *Zaruriy shart:* yechimlarda amallar soni $n \cdot n$ emas, n ta bo'lishi kerak. Yordamchi massivlardan foydalanish mumkin.

Tushuntirish: masalan, 2, 9, 8, 5, 9, 3, 4, 5, 2, massiv almashirishdan keyin 9, 0, 9, 9, 0, 4, 5, 0, 0, ko'rinishda bo'ladi.

255. «Rukzak». Berilgan n predmetdan shundaylarini tanlab olish kerakki, ularning jami og'irligi 30 kg.dan kichik, qiymatlari esa eng katta bo'lsin. Tanlangan predmetlarning jami qiymati chop etilsin.

Aniqrog'i — 2 ta musbat sonli $A(n)$ va $B(n)$ massivlar berilgan. Shunday har xil juftli i_1, i_2, \dots, i_k sonlarni tanlash kerakki, natijada

$$a_{i_1} + a_{i_2} + \dots + a_{i_k} < 30$$

$$b_{i_1} + b_{i_2} + \dots + b_{i_k} = \max$$

bo'lsin, faqat \max miqdorini chop eting.

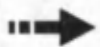
! *Izoh:* predmetlarni a_i — og'irlik, b_i — qiymat, b_i/a_i — baho yoki yana qandaydir bir boshqa belgiga ko'ra o'sib borish yoki kamayib borish tartibida joylashtirilgan, deb hisoblash mumkin.



3.1. QUYI MURAKKABLIKDAGI MASALALAR DASTURLARI



3.1.1. Chiqarish ko'rsatmali dasturlar

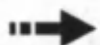


20-masala

begin

```
writeln ('Inson aqli — galaktika yo'lida chiroq,');  
writeln ('Balki kosmos mozorida ko'milar tanim?');  
writeln ('Yo'q-yo'q, inson darichasin ochgan u chog'da,');  
writeln ('Tarixlarga — u, o'lmas, deb yozdi Vatanim.');
```

end.



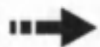
25-masala

uses Crt;

begin

```
TextBackGround (Blue); {tag rangi}  
TextColor (LigthGray); {belgilar rangi}  
ClrScr; {ekranni tozalash}  
writeln ('Dunyo bamisoli bir zarrin ko'za,');  
writeln ('Suvi goh achchig'-u, goh shirin bo'za,');  
writeln ('Ey, inson, umringga bino qo'yma ko'p,');  
writeln ('Qazoyi muallaq turar bosh uzra');  
writeln ;  
writeln ('Pahlavon Mahmud');  
readln; {Ekrandan she'r yo'qolmasligi uchun}
```

end.



26-masala

Uses Crt;

begin

```
TextBackGround (Black);  
ClrScr;  
TextColor (Red);
```

```

write ('Qizil');
TextColor (LigthRed); {olovrangni och qizil bilan almash-
tiramiz}

write ('Och qizil');
TextColor (Yellow);
write ('sariq');
TextColor (Green);
write ('yashil');
TextColor (LigthBlue);
write ('havorang');
TextColor (Blue);
write ('ko'k');
TextColor (Magenta);
write ('binafsharang');
readln; {ifoda ekrandan yo'qolmasligi uchun}
end.

```



3.1.2. Chiziqli tasnifli dasturlar



34-masala

(Ikki tomoni va ular orasidagi burchagiga ko'ra uchburchak yuzasini topish)

```

var
  a,b: real; {tomonlar uzunligi}
  f: real; {graduslarda ifodalovchi burchak kattaligi}
  s: real; {uchburchak yuzi}
begin
  writeln ('Uchburchak yuzini topish');
  writeln ('Uchburchak tomonlari uzunligi (sm):');
  write (' ->');
  readln (a, b);
  writeln ('Uchburchak ikki tomoni o'rtasidagi burchak kattaligi
    (gradus):');
  write (' ->');
  readln [f];
  {s = a*h/2, h (uchburchak balandligi) h=b*sin(f) formula
  bilan hisoblanishi mumkin. Lekin, Turbo Pascalda Sin
  funksiyaning argumenti radianlarda ifodalanishi kerak (1 rad.
  = 180/3.1415925, bu yerda, 3.1415926 — "PI" soni)}
  s:=a*b*sin (f*3.1415926/180) /2;

```

```
writeln;  
writeln ('uchburchak yuzi:' ,s:6:2, 'sm.kv.);  
readln;  
end.
```



35-masala

(Ikki parallel ulangan elementlardan iborat elektr zanjir qarshiligini hisoblash)

var

```
r1,r2: real; {zanjir elementlari qarshiligi};  
r: real; {zanjirning umumiy qarshiligi}
```

begin

```
writeln ('Elementlari parallel ulangan elektr zanjir qarshiligini  
hisoblash')  
writeln ('Boshlang'ich qiymatlarni kiriting:');  
write ('Birinchi qarshilik kattaligi (Om) →');  
readln (r1);  
write ('Ikkinchi qarshilik kattaligi (Om) →');  
readln (r2);  
r: = r1 * r2 / (r1 + r2);  
writeln;  
writeln ('Zanjir qarshiligi:',r:6:2,'Om');  
readln;  
end.
```



36-masala

(Dala hovliga borish va qaytish qiymatlarini hisoblash)

var

```
mas: real; {dala hovligacha bo'lgan masofa}  
sarf: real; {100 km.ga benzin sarfi}  
baho: real; {1 litr benzinning bahosi}  
summ: real; {dala hovliga borish va qaytish qiymati}
```

begin

```
writeln ('Dala hovliga borish va qaytish qiymatini hisoblash');  
write ('Dala hovligacha masofa (km) →');  
readln (mas);  
write ('Benzin sarfi (100 km masofaga, litr) →');  
readln (sarf);  
write ('Litr benzin bahosi (so'm) →');  
readln (baho);
```

```

summ: = 2 * mas / 100 * sarf * baho;
writeln;
writeln ('Dala hovliga borish va qaytish qiymati', summ:6:2,
'so'm.');
```

readln;

end.



37-masala

{Silindr sirt yuzasini hisoblash}

var

r: real; {silindr asosi radiusi}

h: real; {silindr balandligi}

s: real; {silindr sirt yuzasi}

begin

writeln ('Silindr sirt yuzasini hisoblash');

writeln ('Boshlang'ich qiymatlar:');

write ('Asos radiusi (sm) →');

readln (r);

write ('Silindr balandligi (sm) →');

readln (h);

{pi — nomlangan o'zgarma, uning qiymati "PI" soniga teng. Sqr — Turbo Pascal funksiyasi, uning qiymati argument kvadratiga teng, ya'ni $Sqr(x) = x^2$ }

s: = 2 * pi * sqr(r) + 2 * pi * r * h;

writeln;

writeln ('Silindr sirt yuzasi', s:6:2,'sm.kv.');

readln;

end.



41-masala

(Verstlardagi masofani kilometrlarda hisoblash)

var

v: real; {verstlardagi masofa}

k: real; {kilometrlardagi masofa}

begin

writeln ('Masofani Verstlardan kilometrlarga aylantirish);

writeln ('Verstlardagi masofani kiriting va <Enter>ni bosing');

write ('→');

readln (v);

k: = v * 1.0668;


```
writeln (v:6:2,'verst – bu', k:6:2,'km');
writeln ('Dastur ishini tugatish uchun <Enter>ni bosing');
readln;
end.
```



3.1.3. IF tanlov ko'rsatmali dasturlar



43-masala

{Omonatlarga ko'ra daromadni hisoblash}

var

```
sum: real; {jami omonat}
muddat: real; {omonat muddati}
foiz: real; {qo'shimcha foiz}
daromad: real; {omonatga ko'ra daromad}
```

begin

```
writeln ('Omonatga ko'ra daromadni hisoblash. ');
writeln ('Boshlang'ich qiymatlarni kiriting. ');
write ('Omonat miqdori (so'm.) → ');
readln (sum);
write ('Omonat muddati (kunlarda) → ');
readln (muddat);
write ('qo'shimcha foiz (yillik) → ');
readln (foiz);
daromad: = (sum * foiz / 100) / 365 * muddat;
           {365 – bir yildagi kunlar soni}
sum: = sum + daromad;
writeln;
writeln ('-----');
writeln ('Daromad:', daromad:9:2,'so'm. ');
writeln ('Omonat muddati oxiridagi yig'indi:',sum:9:2,
        'so'm. ');
```

readln;

end.



44-masala

{Minutlarda berilgan kattalikni unga mos keluvchi soat va minutlarda ifodalangan qiymatga aylantirish}

var

```
min: integer; {minutlarda ifodalangan oraliq}
t: integer; {soat miqdori}
```

```

    m: integer; {minut miqdori}
begin
    writeln ('Vaqt oralig'i miqdorini kiriting (minutlarda) va
            <Enter>ni bosing');
    write ('→ ');
    readln (min);
    t:= min div 60;
    m:= min mod 60;
    writeln;
    writeln (min, 'min. — bu', t,'soat.',m,'min.');
```

```

end.
```



45-masala

{Sonni pul formatida ifodalash}

```

var
    n: real; {kasr son}
    r: integer; {sonning butun qismi (so'm)}
    k: integer; {sonning kasr qismi (tiyin)}
begin
    writeln ('Sonni pul formatiga o'zgartirish.').
    write ('Kasr sonini kiriting → ');
    readln (n);
    r:= Round (n*100) div 100;
    k:= Round (n*100) mod 100;
    writeln;
    writeln (n:6:2,' so'm. - bu ',r,' so'm.',k,' tiyin.');
```

```

end.
```



46-masala

{Bo'linmani hisoblash}

```

var
    a,b,c: real; {bo'linuvchi, bo'luvchi va bo'linma}
begin
    writeln ('Bo'linmani hisoblash');
    writeln ('Bitta satrdan bo'linuvchi va bo'luvchini kiriting,');
    writeln ('keyin <Enter>ni bosing');
    write ('→ ');
    readln (a, b);
    if b <> 0 then
```

```

begin
  c:=a/b;
  writeln (a ' a:6:3'ni ' b:6:3 ' ga bo'lishdan olingan bo'linma:
           'c:6:3 'ga teng);
end
  else
  writeln ('Xato! Bo'luvchi nolga teng bo'lmasligi kerak!');
  readln;
end.

```



47-masala

{Halqa yuzasini hisoblash}

```

var
  r1,r2 : real; {halqa va teshik radiusi}
  s : real;     {halqa yuzasi}
begin
  writeln ('Halqa yuzini hisoblash:');
  writeln ('Boshlang'ich qiymatlarni kiriting:');
  write ('halqa radiusi (sm)→');
  readln (r1);
  write ('teshik radiusi (sm)→');
  readln (r2);
  if r1 > r2 then
    begin
      s:=2*3.14 * (r1-r2);
      writeln ('Halqa yuzasi ', s:6:2,'kv.sm');
    end
    else writeln ('Xato! Teshik radiusi halqa radiusidan katta
                 bo'lmasligi kerak');
      writeln ('Dastur ishini tugatish uchun <Enter>ni bosing');
      readln;
    end.

```



48-masala

{Elektr zanjir qarshiligini hisoblash}

```

var
  r1,r2; real; {zanjir qarshiligi kattaligi}
  n: real; {yig'indi qarshilik}
  t: integer; {elementlarni ulash turi:
              1-ketma-ket;
              2-parallel}

```

begin

writeln ('Elektr zanjir qarshiligini hisoblash');

writeln ('Boshlang'ich qiymatlarni kiriting:');

write ('Birinci qarshilik miqdori (Om)→');

readln (r1);

write ('Ikkinchi qarshilik miqdori (Om)→');

readln (r2);

write ('Elementlarni ulash turi (1-ketma-ket, 2-parallel) →');

readln (t);

if t=1 *then* r := r1+r2

else r := r1*r2/(r1+r2);

writeln ('Zanjir qarshiligi: ',r:5:2, 'Om');

readln;

end



49-masala

{Kvadrat tenglamani yechish}

var

a,b,c:real; {tenglama koeffitsientlari}

x1,x2:real; {tenglama ildizlari}

d:real; {diskriminant}

begin

writeln ('*Kvadrat tenglamani yechish*');

write ('Bitta satrdan koeffitsientlar qiymatini kiriting');

writeln ('va <Enter>ni bosing');

write ('→');

readln (a,b,c); {koeffitsientlarni kiritish}

d := $b*b - 4*a*c$; {diskriminantni hisoblash}

if *d* >= 0

then

begin

x1 := $-b + \text{sqrt}(d)/(2*a)$;

x2 := $-b - \text{sqrt}(d)/(2*a)$;

writeln ('Tenglama ildizlari:');

writeln ('x1 =',x1:9:3);

writeln ('x2 =',x2:9:3);

end

else

writeln ('Haqiqiy ildizlar yo'q');

end.



50-masala

{Imtiyozni hisobga olgan holda xarid qiymatini hisoblash}

```
var
  summ: real; {xarid jami}
begin
  writeln ('Imtiyozni hisobga olgan holda xarid qiymatini
    hisoblash');
  writeln ('Xarid qiymatini kiriting va <Enter>ni bosing. ');
  write ('→');
  readln (summ);
  if summ > 5000
    then {imtiyoz beriladi}
      begin
        if summ > 10000
          then
            begin
              writeln ('Sizga 5 % imtiyoz beriladi');
              summ:=0.95*so'm;
            end
          else
            begin
              writeln ('3 % imtiyoz beriladi');
              summ: = 0.97 * so'm;
            end;
          writeln ('imtiyozni hisobga olgan holda xarid jami:
            summ: 6:2, 'so'm. ');
        end
      else
        writeln ('Imtiyoz berilmaydi');
  readln;
end.
```



52-masala

{Tarixdan bilimni tekshirish}

```
var
  jav: integer; {sinaluvchining javobi}
begin
  writeln ('Alisher Navoiy nechanchi yil tug'ilgan?')
  writeln ('Sonni kiriting va <Enter>ni bosing');
  write ('→');
```

```

readln (jav);
if jav = 1441
    then writeln ('To'g'ri.')
    else
        begin
            write {'Adashdingiz!'};
            writeln ('Alisher Navoiy 1441-yilda tug'ilgan,')
        end;
readln;
end.

```



53-masala

{Fanlar tarixidan bilimni tekshirish}

```

var
    jav: integer; {sinaluvchi tanlagan javob raqami}
begin
    writeln ('Kibernetika fani asoschisi kim?');
    writeln ('1. Andre Mari Amper');
    writeln ('2. Isaak Nyuton');
    writeln ('3. Norbert Viner');
    writeln;
    writeln ('to'g'ri javob raqamini kiriting va <Enter> ni bosing');
    write ('→');
    readln (jav) ;
    if jav = 2
        then writeln ('To'g'ri.')
        else
            begin
                writeln ('Adashdingiz. ');
                writeln ('Kibernetika fani asoschisi Norbert Viner. ');
            end;
    readln;
end.

```



54-masala

{Ikki butun sonni taqqoslash}

```

var
    a,b: integer; {taqqoslanadigan sonlar}
begin
    write ('Bitta satrdan ikkita butun sonni kiriting');

```

```

writeln ('va <Enter>ni bosing');
write ('→');
readln (a, b);
if a = b
    then writeln ('sonlar teng')
    else if a < b
        then writeln (a,'kichik',b)
        else writeln (a,'katta',b);
readln;
end.

```



55-masala

{Sonlarni ko'paytira olishni tekshirish}

```

var
    m1,m2,p: integer; {ko'paytuvchilar va ko'paytma}
    jav: integer; {sinaluvchi javobi}
begin
    Randomize; {tasodifiy sonlar generatorini initsializatsiyalash}
    m1:= Random (9)+1; {Random (a) funksiyasi qiymati, y}
    {0 dan a-1 gacha sohadagi son}
    m2:= Random (9)+1;
    p:= m1 * m2;
    writeln (m1,'x',m2,'necha bo'ladi?');
    writeln ('Javobni kiriting va <Enter>ni bosing');
    write ('→');
    readln (jav);
    if p = jav
        then writeln ('To'g'ri.')
        else writeln ('Siz adashdingiz.', m1, 'x', m2,',p);
    readln;
end.

```



59-masala

{Klaviaturadan kiritilgan sonning juftligini tekshirish}

```

var
    n: integer; {foydalanuvchi kiritgan son}
begin
    writeln ('Butun sonni kiriting va <Enter>ni bosing. ');
    write ('→');

```

```

readln (n);
write ('Son', n, ' — ');
if n mod 2 = 0
  then writeln (n, 'soni juft.')
  else writeln (n, 'soni toq.');
```

readln; {natija ekrandan yo'qolmasligi uchun}

end.



61-masala

{Shanba va yakshanba kunlari beriladigan imtiyozni hisobga olgan holda telefonda so'zlashuv qiymatini hisoblash}

var

```

Time: integer; {so'zlashuv uzunligi}
Day: integer; {hafta kuni}
Summa: real; {so'zlashuv qiymati}
```

begin

```

writeln ('Telefonda so'zlashuv qiymatini hisoblash.');
```

writeln ('Boshlang'ich qiymatlarni kiriting:');

write ('So'zlashuv uzunligi');

write ('(minutlarning butun miqdori)→');

readln (Time);

write ('Hafta kuni');

write ('(1-dushanba, ... , 7-yakshanba)→');

readln (Day);

Summa:= 20.3 * Time; {1 minut bahosi 20.3 so'm.}

if (Day = 6) or (Day = 7)

then begin

writeln ('20 % li imtiyoz beriladi.');

Summa:=Summa * 0.8;

end,

writeln ('So'zlashuv qiymati:', Summa: 8:2, 'so'm.');

readln;

end.



62-masala

{Vaznni nazorat qilish}

var

```

w:real; {vazn}
h:real; {bo'y}
opt: real; {optimal vazn}
```



```

d:real; {optimal vazndan farqlanish}
begin
writeln ('Bir satrdan bo'shliq tashlab kiriting');
writerln ('bo'y (sm) va vazn (kg), keyin <Enter>ni bosing');
  write ('→');
  readln (h,w);
  opt:=h - 100;
  if w = opt
    then
      writeln ('Sizning vazningiz optimal!')
    else
      if w<opt
        then begin
          d:=opt-w;
          writeln ('Sizga', d:5:2, 'kg.ga semirish kerak');
        end
      else
        begin
          d:=w-opt;
          writeln ('Siz', d:5:2, 'kg.ga ozishingiz kerak');
        end;
  readln;
end.

```



3.1.4. CASE ko'rsatmali dasturlar



64-masala

{Oy raqamiga ko'ra yil mavsumini aniqlash}

```

var
  oy: integer; {oy raqami}
begin
  writeln ('Oy raqamini kiriting (1 dan 12 gacha bo'lgan sonlar)
    va <Enter>ni bosing');
  write ('→');
  readln (oy);
  case oy of
    1,2,12:   writeln ('Qish');
    3.. 5:    writeln ('Bahor');
    6.. 8:    writeln ('Yoz');
    9.. 11:   writeln ('Kuz');
    else      writeln ('Son 1 dan 12 gacha bo'lishi kerak');
  end;
end.

```

```
end;  
readln;  
end.
```



65-masala

{Shaharlararo telefonda so'zlashuv qiymatini hisoblash}

var

```
kod: integer; {shahar kodi}  
baho: real; {minut bahosi}  
uzun: integer; {so'zlashuv uzunligi}  
summ: real; {so'zlashuv qiymati}
```

begin

```
writeln ('Telefonda so'zlashuv qiymatini hisoblash.');
```

```
writeln ('Boshlang'ich qiymatlarni kiriting:');
```

```
write ('Shahar kodi → ');
```

```
readln (kod);
```

```
write ('So'zlashuv uzunligi (minutning butun miqdori →)');
```

```
readln (uzun);
```

```
write ('Shahar:');
```

```
case kod of
```

371: begin

```
writeln ('Toshkent');
```

```
baho:= 150;
```

```
end;
```

366: begin

```
writeln ('Samarqand');
```

```
baho:=120;
```

```
end;
```

375: begin

```
writeln ('Qarshi');
```

```
baho:=100;
```

```
end;
```

365: begin

```
writeln ('Buxoro');
```

```
baho:= 130;
```

```
end;
```

```
end;
```

```
summ := baho * uzun;
```

```
writeln ('Minut bahosi:', baho: 6:2, 'so'm.');
```

```
writeln ('So'zlashuv qiymati:', summ: 6:2,'so'm.');
```

```
readln;
```

```
end.
```



66-masala

{Navbatdagi kun sanasini aniqlash}

```
var
```

```
  kun: integer;
```

```
  oy: integer;
```

```
  yil: integer;
```

```
  oxir: boolean; {TRUE, agar joriy kun oyning oxirgi kuni  
                bo'lsa}
```

```
  r: integer,    {agar yil kabisa yili bo'lsa, unda yilni 4 ga bo'-  
                lishdagi qoldiq nolga teng}
```

```
begin
```

```
  write ('Bugungi sanani raqamlarda kiriting');
```

```
  write ('(kun oy yil) →');
```

```
  readln (kun, oy, yil);
```

```
  oxir. = FALSE;
```

```
  case oy of
```

```
    1, 3, 5, 7, 8, 10, 12 : if kun = 31
```

```
                            then oxir. = TRUE;
```

```
    4, 6,9,11 : if kun = 30
```

```
                            then oxir. = TRUE;
```

```
    2 : if kun = 28 then
```

```
        begin
```

```
          r:=yil mod 4;
```

```
          if r <> 0
```

```
            then oxir. = TRUE;
```

```
        end;
```

```
end; {case}
```

```
if oxir
```

```
  then begin
```

```
    writeln ('Oyning oxirgi kuni!');
```

```
    kun:=1;
```

```
    if oy = 12
```

```
      then begin
```

```
        oy:=1;
```

```
        yil:=yil + 1;
```

```
        writeln ('Yangi yilingiz bilan!');
```

```

                end
            else oy:= oy + 1;
        end
    else kun:=kun + 1;
    writeln ('Erta ', kun, ' . ', oy, ' . ', yil);
    readln;
end.

```



3.1.5. FOR ko'rsatmali dasturlar



68-masala

{Birinci o'nta butun musbat sonlar kvadratlari jadvalini chiqarish}

```

var
    x: integer; {son}
    y: integer; {son kvadrati}
    i: integer; {sikli hisobchisi}
begin
    writeln ('Kvadratlar jadvali');
    writeln ('-----');
    writeln ('Son Kvadrat');
    writeln ('-----');
    for i:= 1 to 10 do
        begin
            y:=x*x;
            writeln (x:3,y:10);
            x:=x+1;
        end;
    writeln ('-----');
    readln;
end.

```



69-masala

{Birinci n ta butun musbat sonlar yig'indisini hisoblash}

```

var
    n: integer; {qo'shiladigan sonlar miqdori}
    summ: integer; {yig'indi}
    i: integer; {sikli hisobchisi}
begin

```

```

writeln ('Musbat sonlar yig'indisini hisoblash');
write ('Qo'shiladigan sonlar miqdorini kiriting —>');
readln (n);
summ: = 0;
for i:=1 to n do
    summ: = summ + i;
write ('Birinchi', n, 'butun musbat sonlar yig'indisi');
writeln ('teng', summ);
readln;
end.

```



72-masala

{1,3,5,7 ... qator xususiy yig'indisini hisoblash}

```

var
e: integer; {qator hadi}
n: integer; {qo'shiladigan hadlar miqdori}
summ: integer; {qator hadlari yig'indisi}
i: integer; {siki hisobchisi}
begin
writeln ('1,3,5,7, ... qator xususiy yig'indisini hisoblash');
writeln ('qo'shiladigan hadlar miqdorini kiriting —>');
readln (n);
e:=1; {qator birinchi hadi}
summ: = 0;
for i:=1 to n do
    begin
        summ: = sum + e;    {yig'indiga qator navbatdagi
                             hadini qo'shing}
        e: = e + 2;        {keyingi had qiymatini
                             hisoblang}
    end;
writeln ('qatorning birinchi', n, 'hadlar yig'indisi teng:', summ);
readln;
end.

```



73-masala

{1 + 1/2 + 1/3 - ... qator hadi yig'indisini hisoblash}

```

var
n: integer; {qatorning qo'shiladigan hadlar miqdori}
i: integer; {qator elementi tartib raqami}

```

elem: real; {qator elementi qiymati}
summ: real; {qator hadlari yig'indisi}

begin

writeln ('1+1/2+1/Z + ...qator xususiy yig'indisini hisoblash');

writeln ('qatorning qo'shiladigan hadlar miqdorini kiriting');

write ('→');

readln (n);

summ:=0;

for i:= 1 to n *do*

begin

elem:= 1/i;

summ = *summ* + *elem*;

end;

write ('Birinchi', n, 'had yig'indisi:');

writeln (*summ*:6:4);

readln;

end.



74-masala

{Ikkining darajalar jadvali}

var

n: integer; {daraja ko'rsatkichi}

x: integer; {n darajadagi 2 qiymati}

begin

writeln ('Ikkining darajalar jadvali');

x = 1;

for n:=0 to 10 *do*

begin

writeln (n:2, x:6);

x:=*x**2;

end;

readln;

end.



79-masala

{Funksiyaning jadvali}

const

B = -2; {argument o'zgarish sohasining quyi chegarasi}

A = 2; {argument o'zgarish sohasining yuqori chegarasi}

Dx = 0.5; {argument orttirmasi}

```

var
  x, y: real; {argument va funksiya qiymati}
  n: integer; {nuqtalar soni}
  i: integer; {sikli hisobchisi}
begin
  n:=Round ((A - B)/DX) + 1;
  x:=B;
  writeln ('-----');
  writeln ('          x      |      y' );
  writeln ('-----');
  for i:=1 to n do
    begin
      y:= - 2.4*x*x+5*x-3;
      writeln (x:8:2, ' | ', y:8:2);
      x:=x+DX;
    end;
  writeln ('-----');
  readln;
end.

```



82-masala

{Klaviaturadan kiritiladigan kasr sonlar ketma-ketligining o'rta arifmetikligini hisoblash}

```

const
  L=5; {ketma-ketlik uzunligi}
var
  a: real; {son}
  n: integer; {kiritilgan sonlar miqdori}
  sum: real; {kiritilgan sonlar yig'indisi}
  o'rta: real; {kiritilgan sonlarning o'rta arifmetikligi}
begin
  writeln ('Kasr sonlar ketma-ketligini qayta ishlash');
  writeln ('Har bir son kiritilgandan keyin <Enter>ni bosing');
  sum:=0;
  for n:= 1 to L do
    begin
      write ('→');
      readln (a);
      sum:=sum + a;
    end;
end.

```

```

o'rt:=sum/n;
writeln ('kiritilgan son' n:3, 'Yig'indi:', sum: 6:2);
writeln ('O'rt arifmetik:', o'rt:6:2);
end;
writeln ('Dastur ishini tugatish uchun <Enter>ni bosing');
readln;
end.

```



83-masala

{O'rt arifmetikligini hisoblash va klaviaturadan kiritiladigan kasr sonlar ketma-ketligidan eng kichik va eng katta sonlarni aniqlash}

var

<i>a</i> : real;	{navbatdagi son}
<i>n</i> ; integer;	{sonlar miqdori}
<i>sum</i> : real;	{kiritilgan sonlar yig'indisi}
<i>o'rt</i> : real;	{o'rt arifmetik}
<i>min</i> : real;	{ketma-ketlikning eng kichik soni}
<i>max</i> : real;	{ketma-ketlikning eng katta soni}
<i>i</i> : integer;	{siki hisobchisi}

begin

```

writeln ('Kasr sonlar ketma-ketligini qayta ishlash. ');
write ('Ketma-ketlikdagi sonlar miqdorini kiriting ->');
readln (n);
writeln ('Ketma-ketlikni kiriting. ');
writeln ('Har bir son kiritilganidan keyin <Enter>ni bosing');
write ('->');
readln (a); {ketma-ketlikning birinchi sonini kiritamiz}
           {faraz qilamiz:}
min:=a; {birinchi son eng kichik}
max:=a; {birinchi son eng katta}
sum:=a;
{qolgan sonlarni kiritamiz}
for i:= 1 to n-1 do
  begin
    write ('-> ');
    readln (a);
    sum:=sum + a;
    if a < min then min:=a;
    if a > max then max:=a;
  end;

```

end;


```

o'rta: = sum/n;
writeln ('Sonlar miqdori:', n);
writeln ('O'rta arifmetik:', o'rta:6:2);
writeln ('Minimal son', min:6:2);
writeln ('Maksimal son:', max:6:2);
writeln ('Dastur ishini tugatish uchun <Enter>ni bosing');
readln;
end.

```



84-masala

{Tasodifiy sonlarning uchta ketma-ketligini hosil qiladi va har bir ketma-ketlikning o'rta arifmetikligini hisoblaydi}

```

const
  n=3; {ketma-ketlik miqdori}
  l=10; {ketma-ketlik uzunligi}
var
  r: integer; {tasodifiy son}
  sum: integer; {ketma-ketlik sonlari yig'indisi}
  o'rta: real; {o'rta arifmetik}
  i,j: integer; {sikl hisobchisi}
begin
  writeln ('Tasodifiy sonlar');
  Randomize; {tasodifiy sonlar generatorini initsializatsiyalash}
  for i:=1 to n do
begin
  {ketma-ketlik hosil qilish}
  sum:=0; {nollashtirish!}
  for j:= 1 to l do
begin
  r:=Random (10)+1;
  write (r:3);
  sum:=sum+r;
end
  o'rta:=sum/l;
  writeln ('O'rta arifmetik.:', o'rta:6:2);
end;
writeln ('Dastur ishini tugatish uchun <Enter>ni bosing');
readln;
end.

```



85-masala

 $y=|x|$ funksiya jadvali}

const

 $A = -4;$ {argument o'zgarish sohasining quyi chegarasi} $B = 4;$ {argument o'zgarish sohasining yuqori chegarasi} $DX = -5;$ {argument orttirmasi}

var

 $x, y:$ real; {argument va funksiya qiymati} $n:$ integer; {nuqtalar miqdori} $j:$ integer; {sikl hisobchisi}

begin

writeln (' $y=|x|$ funksiya qiymatlar jadvali'); $n := \text{Round}((B - A)/DX) + 1;$ $x := A;$ for $i := 1$ to n do

begin

 $y := \text{Abs}(x);$ writeln ($x:6:2, y:6:2$); $x := x + DX;$

end;

readln;

end.



86-masala

{7 ga ko'paytirish jadvalini chiqarish}

var

 $m:$ integer; {ko'paytirish jadvali chiqariladigan son (ko'payuvchi)} $n:$ integer; {ko'paytuvchi} $p:$ integer; {ko'paytma}

begin

 $m := 7;$ for $n := 1$ to 10 do

begin

 $p := m * n;$ writeln ($m, 'x', n, ' = ', p$);

end;

readln; {natija ekrandan yo'qolmasligi uchun}

end.



87-masala

{O'nlik sonni ikkilik songa aylantirish}

var

o'n: integer; {o'nlik son}

v: integer; {shakllanadigan raqam xonasi (razryadi) salmog'i}

i: integer; {shakllanadigan raqam xonasi (razryad) tartibi
qiymati (nomeri)}

begin

writeln ('O'nlik sonni ikkilikka aylantirish');

writeln ('0 dan 255 gacha bo'lgan sonni kiriting va <Enter>ni
bosing');

writeln ('→');

readln (o'n);

write (O'n, 'o'nlik soniga ikkilik mos keladi');

v:=128; {bosh razryad salmog'i}

for i:= 1 to 8 do

begin

if o'n >= v then

begin

write ('1');

o'n := o'n - v;

end

else write ('0');

v:= Round (v/2); {navbatdagi razryad salmog'i joriy razryad
salmog'idan ikki marta kam}

end;

readln;

end.



88-masala

{Ekkranga Pifagor ko'paytirish jadvalini chiqarish}

var

i,j: integer; {jadval satr va ustun raqami}

p: integer; {i ning j ga ko'paytmasi}

begin

write (' ': 4); {jadval yuqori chap burchak katagi}

for j:= 1 to 10 do {ustun raqami birinchi satri}

write (j:4);

```

writeln;
for i: = 1 to 10 do
  begin
    write (i:4);           {satr raqami}
    for j: = 1 to 10 do   {jadval satri}
      write (i*j:4);
    writeln;
  end;
readln; {natija ekrandan yo'qolmasligi uchun}
end.

```



89-masala

{1 - 1/3 + 1/5 - 1/7 + . . . qator yig'indisini hisoblash}

```

var
  x: real;           {qator hadi}
  n: integer;       {qo'shiladigan hadlar miqdori}
  summ: real;       {xususiy yig'indi}
  i: integer;       {siki hisobchisi}
begin
  writeln ('1 - 1/3 + 1/5 - 1/7 + ... qator yig'indisini hisoblash');
  write ('qatorning qo'shiladigan hadlar miqdorini kiriting →');
  readln (n);
  for i:= 1 to n do
    begin
      x:= 1/(2*i - 1);
      if (i mod 2) = 0
        then x:= - 1*x;
      summ:= summ + x;
    end;
  writeln ('Qator yig'indisi:', summ:B:6);
  writeln (pi/4:8:6, ' — pi/4 qiymati'); {pi — nomlangan
                                          o'zgarmas}

  readln;
end.

```



90-masala

{Ko'paytirish jadvalini bilishni tekshirish}

```

uses
  Crt;

```

var

son1, son2: integer; {birinchi va ikkinchi son}
nat:integer; {birinchi sonni ikkinchi songa ko'paytirish natijasi}
jav:integer; {Sinaluvchi javobi}
miqdor:integer; {To'g'ri javoblar miqdori}
i:integer;

begin

ClrScr; {ekranni tozalash}
*writeln ('***Ko'paytirish jadvalini bilishni tekshirish***');*
writeln ('Misoldan keyin javobni kiriting va <Enter>ni bosing');
writeln;
miqdor:=0; {to'g'ri javoblar}
Randomize;{tasodifiy sonlar generatorini initsializatsiyalash}
for i: =1 to 10 do {10 ta misol}
begin

son1: = Random(9)+1;
son2: = Random (9)+1;
*nat:=son1 * son2;*
write (' ', son1, 'x', son2, '=');
readln (jav);
if jav = nat
then
miqdor:= miqdor +1
else begin
writeln ('Adashdingiz!', son1, '', son2, '= ', nat);*
end;

end;

writeln ('To'g'ri javoblar:', miqdor);
write ('Sizning bahongiz:');

case miqdor of

10: writeln ('5');
9,8: writeln ('4');
7,6: writeln ('3');
0..5:writeln ('2');

end;

writeln ('Dastur ishini tugatish uchun <Enter>ni bosing');
readln;

end.

*{Sonlarni qo'shish va ayirishni tekshirish}*

uses

Crt;

var

son1, son2:integer; {sonlar}

amal:integer; {Sonlar ustida amallar: 0 — qo'shish,
1 — ayirish}

belgi: char; {amal belgilari — «plyus» yoki «minus»}

nat:integer; {Natija}

jav:integer; {Sinaluvchi javobi}

miqdor:integer; {To'g'ri javoblar miqdori}

buf:integer; {son1 < son2 bo'lgan holda son1 va son 2 ni
almashtirish uchun bufer}

i: integer; {sikl hisobchisi}

begin

ClrScr;

writeln ('Sonlarni qo'shish va ayira bilishni tekshirish.');

writeln ('Misoldan keyin javobni kiriting va <Enter>ni
bosing.');

miqdor:=0;

Randomize;

for i:=1 to 10 do

begin

son1:=Random (9)+1 {birinchi son}

son2:=Random (9)+1 {ikkinchi son}

amal:=Random (2) {sonlar ustida amallar}

if amal=0 then

begin {qo'shish}

nat:=son1 + son2;

belgi:='+';

end

else

begin {ayirish}

belgi:='-';

if son1 < son then

begin {son1 va son2 ni almashtiramiz}

buf:=son2;

son2:=son1;

son1:=buf;

```

                end;
            nat := son1 - son2;
        end;
    write (' ', son1, belgi, son2, '='); {misolni chiqarish}
    readln (jav); {sinovchi javobini hosil qiling}
    if jav = nat
        then
            miqdor := miqdor + 1
        else
            begin
                writeln ('Adashdingiz!', son1, belgi, son2, '=', nat);
            end;
        end;
    writeln ('To'g'ri javoblar:', miqdor);
    write ('Bahongiz:');
    case miqdor of
        10: writeln ('5');
        9,6: writeln ('4');
        7,6: writeln ('3');
        0..5: writeln ('2');
    end;
    writeln ('Dastur ishini tugatish uchun <Enter>ni bosing');
    readln;
end.

```



92-masala

{Oddiy elektron soatlar}

uses Crt; {Crt kutubxonasi protsedura va funksiyalaridan foydalanish}

var

min, sec: integer; {minutlar, sekundlar}

begin

ClrScr; {ekranni tozalash}

for min := 1 to 3 do

begin

for sec := 1 to 60 do

begin

Delay (1000); {1000 ms to'xtatib turish}

GotoXY (1,1); {kursorni 1-satrning 1-ustuniga o'rnatish}

```

write (min, ':', sec, ' ');
if KeyPressed {agar klavish bosilgan bo'lsa}
then Halt;      {dasturni tugatish}
end;
end;
end.

```



3.1.6. REPEAT ko'rsatmali dasturlar



97-masala

{Klaviaturadan kiritilgan butun sonning tub sonligini tekshirish}

```

var
n: integer;
d: integer;
r: integer;
begin
write ('Butun sonni kiriting →');
readln(n);
d:= 2; {avval ikkiga bo'lamiz}
repeat
r:=n mod d;
if r<>0 {n d ga qoldiqsiz bo'linmadi}
then d:=d + 1;
until r=0; {hozircha n bo'linadigan sonni topmadik}
if d=n
then writeln (n, ' — tub son.')
else writeln (n, ' — tub son emas. ');
end.

```

98-masala

{«Sonni top» o'yini}

```

const
urinish = 5;      {o'yinchiga beriladigan urinishlar soni}
var
comp: integer;    {kompyuter «o'ylagan» son}
o'yinchi: integer; {o'yinchi varianti}
n: integer;       {o'yinchining urinishlar soni}
begin
Randomize; {tasodifiy sonlar generatorini initsializatsiyalash}

```



```

comp:=Random (9)+1; {kompyuter son o'yladi}
writeln ('«Sonni top» o'yini');
writeln ('Kompyuter 1 dan 10 gacha son «o'yladi»');
writeln ('uni', urinish, 'urinishda toping. ');
writeln ('Sonni kiriting va <Enter>ni bosing. ');
repeat
    n: = n + 1;
    write ('→ ');
    readln (o'yinchi);
until (n = urinish) or (comp = o'yinchi);
if comp = o'yinchi
    then writeln ('Yutdingiz!')
    else writeln ('Yutqazdingiz! Kompyuter', comp, 'sonini
        o'ylagan edi');

    readln;
end.

```



99-masala

{Taymer}

```

uses Crt;
var
    min, sec: integer;      {to'xtatish: minut va sekund}
begin
    writeln ('Minut va sekundlarda to'xtatish miqdorini kiriting');
    writeln ('Masalan, 2 30 va <Enter>ni bosing. ');
    write ('→');
    readln (min, sec);
    ClrScr;
    GotoXY(2,2);
    write (min, ':', sec, ' ');
    repeat
        if sec = 0 then
            begin
                min:=min - 1;
                sec: =60;
            end;
        repeat
            Delay (1000);    {to'xtatish 1 sek}
            sec: = sec - 1;
            GotoXY(2,2);

```

```

write (min, ':', sec, ' ');
if KeyPressed {agar klavish bosilgan bo'lsa}
then Halt; {dasturni yakunlang}
until sec =0;
until (min = 0) and (sec = 0);
{tovushli signali}
Sound (1000); {chastotasi 1000 gersga teng bo'lgan tovush
signalni qo'shing}
Delay (500);
Nosound; {tovushni o'chiring}
end.

```



3.1.7. WHILE ko'rsatmali dasturlar



103-masala

{PI sonini hisoblash}

```

var
p:real; {P/ning hisoblanadigan qiymati}
a:real; {hisoblash aniqligi}
n:integer; {qator hadi raqami}
elem: real; {qator hadi qiymati}
begin
p:=0;
n:=1;
elem:=1; {boshlang'ich qiymat}
write ('PI hisoblash aniqligini bering →');
readln (a);
writeln ('P/ni quyidagi aniqlik bilan hisoblang', a:9:6);
while elem >= a do
begin
elem:=1/(2*n-1);
if (n MOD 2) = 0
then p:=p-elem
else p:=p+elem;
n:=n+1;
end;
p:=p*4;
writeln (a:9:6, 'aniqlikdagi PI qiymati', p:9:6, 'ga teng');
writeln ('Qatorning, 'n', hadi qo'shilgan. ');
readln;
end.

```

104-masala

{Ikki butun sonning eng katta umumiy bo'luvchisini topish}

var

n1, n2:integer; {EKUB topilayotgan sonlar}
ekub:integer; {eng katta umumiy bo'luvchi}
r:integer; {n1 ni n2 ga bo'lishdagi qoldiq}

begin

writeln ('Ikki butun son uchun eng katta umumiy bo'luvchini hisoblash');

write ('Bir qatordan ikkita sonni kiriting');

writeln ('va <Enter>ni bosing');

write ('→');

readln (*n1*, *n2*);

while (*n1 mod n2*) <>0 do

begin

r:=*n1 mod n2*; {bo'lishdagi qoldiq}

n1:=*n2*;

n2:= *r*;

end;

ekub:=*n2*;

writeln (*n1*, ' va ', *n2*, ' sonlarining EKUB : ', *ekub*);

readln;

end.



3.1.8. Belgilar va qatorlar ishlatilgan dasturlar

109-masala

{Salomlashish}

var

ism: string [40]; {foydalanuvchi ismi}

begin

writeln ('Ismingiz?');

writeln ('Ismingizni kiriting va <Enter>ni bosing');

write ('→');

readln (*ism*);

writeln ('Salom! ', *ism*, ' ! ');

readln;

end.



110-masala

{Axborotni belgilar bo'yicha chiqarish}

```
uses
  crt;          {Delay protsedurasini ishlatish uchun}
var
  msg: string [80];    {axborot}
  n; integer;         {chiqariladigan belgi raqami}
begin
  msg:= 'Buyuk dasturchiga salom!';
  for n:=1 to Length (msg) do
    begin
      write (msg[n]);
      Delay (100);    {0.1 sekundga to'xtatish}
    end;
  readln;
end.
```



111-masala

{Kiritilgan belgi kodini ASCII standartida chiqarish}

```
var
  belgi; char;    {belgi}
  code: integer; {belgi kodi}
begin
  writeln ('belgini kiriting va <Enter>ni bosing. ');
  writeln ('Dastur ishini tugatish uchun nuqtani kiriting. ');
  repeat
    write ('→');
    readln (belgi);
    code:= Ord (belgi);
    writeln ('Belgi:', belgi, 'Kod:', code);
  until belgi= '.';
end.
```



112-masala

{Belgilarni kodlashtirish jadvalini chiqarish}

```
var
  ch:char; {belgi}
  o'n: integer, {belgining o'nlik kodi}
  ij: integer;
```

```

begin
  o'n:=0;
  for i:=0 to 15 do {o'n oltita satr}
    begin
      o'n:=i;   {128–255 kodlar belgilari uchun
                kodlashtirish jadvalini hosil qilishda bu
                ko'rsatmani o'n:=i+128 ga almashtirish
                kerak;}
      for j:=1 to 8 do {sakkizta ustun}
        begin
          if (o'n < 7) or (o'n >=14)
            then
              write (o'n:4, '-';
                    chr (o'n):1, chr (179))
            else {CR, LF, TAB belgilari ifodalanmaydi}
              write (o'n:4, '- ', chr (179));
          o'n:= o'n +16;
        end;
      writeln {ekranning yangi satriga o'tish};
    end;
  readln;
end.

```

113-masala

```

var
  satr: string [80];      {matn satriga}
  uzun: integer;         {satr uzunligi}
  i: integer;            {qayta ishlanadigan belgi raqami}
begin
  writeln ('Matn satrini kiriting va <Enter>ni bosing');
  writeln ('→');
  readln (satr);
  uzun:=Length (satr);
  for i:=1 to uzun do
    case satr [i] of
      'a' . . 'n': satr [i]:= chr (ord (satr [i]) - 32);
      'p' . . 'ya': satr [i]:= chr (ord (satr [i]) - 80);
    end;
  writeln ('Yuqori registrga aylantirilgan satr:'),
  writeln (satr);
  readln;
end.

```

114-masala

{Satrning boshlang'ich bo'shliqlarini olib tashlash}

```
var
  satr: string [80];      {satr}
begin
  writeln ('Satrning boshlang'ich bo'shliqlarini olib tashlash. ');
  write ('Satrni kiriting:');
  readln (satr);
  while (pos(' ', satr) = 1) and (length (satr) > 0) do
    delete (satr, 1, 1);
  write ('Boshlang'ich bo'shliqlarsiz satr:', satr);
  readln;
end.
```

115-masala

{Klaviaturadan kiritilgan sonning butunligini tekshirish}

```
var
  satr: string [40]; {satr}
  n: integer;      {tekshiriladigan belgi raqami}
begin
  writeln ('Sonni kiriting va <Enter>ni bosing');
  write ('→');
  readln (satr);
  n:=1;
  while (n <= Length(satr)) and
    ((satr [n] >= '0') and (satr [n] <= '9'))
    do n:=n+1;
  write ('Kiritilgan satr');
  if n < Length (satr)
    then write ('butun son bo'lmaydi');
  writeln ('butun son bo'ladi. ');
  readln;
end.
```

116-masala

{Kiritilgan satrning o'n oltilik son ekanligini tekshirish}

```
var
  satr: string [20]; {satr}
  i: integer;      {tekshirilayotgan belgi raqami}
```

```

    error: boolean; {satrda xato belgi bor}
begin
    writeln ('o'n oltilik sonni kiriting va <Enter>ni bosing');
    write ('→');
    readln (satr);
    {kiritilgan satrni yuqori registrga almashtiramiz}
    for i:=1 to Length (satr)
        do satr [i] :=UpCase (satr [i]);
    i:=1;
    error:=FALSE;
    while (i <= Length (satr)) and (not error)
        do if ((satr [i] >= '0') and (satr [i] <= '9')) or
            ((satr [i] >= 'A') and [satr [i] <= 'F'))
            then i:= i+1
            else error:=TRUE;
        write ('satr');
        if error
            then write ('o'n oltilik son bo'lmaydi');
        writeln ('o'n oltilik son bo'ladi. ');
        readln;
end.

```

117-masala

{Kiritilgan satr ishorasiz kasr son ekanligini tekshiradi}

```

var
    satr: string [20]; {satr}
    i: integer; {tekshirilayotgan belgi raqami}
    err: boolean; {TRUE — satr kasr son emas}
begin
    writeln ('Kasr sonni kiriting va <Enter>ni bosing');
    write ('→');
    readln (satr);
    i:=1;
    err:=TRUE; {satr kasr son bo'lmasin}
    if (satr [i] >='1') and (satr [i] <= '9') then {raqamning birinchi
                                                    belgisi}
begin
    {raqamdan keyin yana raqam kelishi mumkin}
    while (satr [i] >='1') and (satr [i] <= '9') and (i<Length(satr))
        do i:= i +1;

```

```

{raqamlardan keyin nuqta keladi, lekin u oxirgi belgi emas}
if ((satr [i] = ' . ') and (i < Length(satr))) then {nuqta}
  begin
    i:= i + 1;
    {nuqtadan keyin hech bo'lmaganda bitta raqam
    bo'lishi kerak}
    if ((satr [i] >='1') and [satr [i] <='9']) then
      begin
        while ((satr [i] >='1') and (satr [i] <='9') and
        (i<Length(satr)))
          do i:= i + 1;
        if i= Length(satr) {raqamning oxirgi belgisi}
          then err:=FALSE; {xato to'g'risidagi faraz yolg'on}
          end;
        end;
      end;
    write ('Satr');
    if err
      then write ('kasr son bo'lmaydi');
      writeln ('kasr son bo'ladi. ');
      readln;
  end.

```



118-masala

{Klaviaturadan kiritilgan sakkiz xonali ikkilik sonni o'nlikka aylantirish}

var

```

ikki: string [8];   {ikkilik sonini ifodalash}
o'n: integer;      {o'nlik son}
r: string [1];     {ikkilik sonining i-xonasi}
s: integer;        {ikkilik soning i-xona salmog'i}
i: integer;        {ikkilik son xonasining raqami}

```

begin

```

writeln ('Sakkiz xonalik ikkilik sonini kiriting');
writeln ('va <Enter>ni bosing. ');
write ('→');
readln (ikki);
if Length (ikki) <> 8 then
  writeln ('Son sakkiz xonali bo'lishi kerak.')
  else begin
    o'n := 0;

```



```

s:=128; {bosh (8) xonali ikkilik sonining
        salmog'i}
for i:=1 to 8 do
  begin
    r:=ikkil[i]; {i — xonani ajrating}
    if r = '1' then
      o'n:=o'n +s;
    s:=Round (s/2) ; {navbatdagi xona
                    salmog'ini hisoblang}
  end;
write (ikki, 'Ikkilik soniga ', o'n, 'o'nlik son
      mos keladi');
end;
writeln ('Dastur ishini tugatish uchun <Enter>ni bosing. ');
readln;

```

end.



119-masala

```

var
  satr: string [2]; {o'n oltilik son}
  d: integer;       {kiritilgan o'n oltilik songa mos keluvchi
                    o'nlik son}
  s: integer;       {o'n oltilik son xonasining salmog'i}
  i: integer;
begin
  writeln ('ikki xonali o'n oltilik sonni kiriting', 'va <Enter>ni
          bosing. ');
  write ('→');
  readln (satr);
  {kiritilgan satrni yuqori registrga aylantiramiz}
  for i:=1 to Length (satr)
    do satr [i]: = UpCase (satr [i]);
  i = Length (satr); {kichik xonadan qayta ishlaymiz}
  s: = 1;           {va uning vazni 1 ga teng bo'ladi}
  while (i > 0) and
    (((satr [i] >='0') and (satr [i] <= '9')) or
     ((satr [i] >= 'A') and (satr [i] <= 'F'))) do
begin
  {bu yerda, belgi raqam yoki A dan F gacha lotin harflari}
  if (satr [i] >= '0') and (satr [i] <= '9')
    then d:=d+s*(Ord (satr [i] - 48)
        {Ord ('0') = 48, Ord ('1') = 49, va h.k.}
    else d:=d+s*(Ord (satr [i] - 55));

```

```

        {Ord ('A') = 65, Ord ('B') = 66, va h.k.}
        i:= i - 1;    {oldingi xonaga}
        s:=s*16;
    end;
    if i=0          {hamma xonalar ishlab chiqilgan}
        then writeln (satr, 'O'n oltilik songa o'nlik', d, 'soni
                        mos keladi')
        else writeln ('Kiritilgan satr o'n oltilik raqam
                        bo'lmaydi.');
```

readln;

end.

120-masala

*{O'nlik sonni foydalanuvchi ko'rsatgan sanoq tizimiga
(ikkilikdan o'nlikkacha) o'tkazish}*

```

uses Crt;
var
    a:integer;    {Sanoq tizimi asosi}
    n:integer;    {Boshlang'ich son}
    r:integer;    {Sonni sanoq tizimi asosiga bo'lgandagi chiqqan
                  qoldiq.}
    s:string [16]; {Sonni berilgan sanoq tizimida ifodalash.}
    buf:string [1];
begin
    ClrScr;
    Write ('Butun sonni kiriting →');
    readln (n);
    write ('Sanoq tizimi asosini kiriting →');
    readln (a);
    {boshlang'ich sonni sanoq tizimi asosiga bo'lishdan chiqqan
    qoldiqni tizimi asosidan katta bo'lgunga qadar bo'lamiz.
    Har bir qadamdagi bo'lish qoldig'i — navbatdagi son}
    repeat
        r:=n mod a; {navbatdagi raqam}
        n:=n div a; {bo'lishning butun qismi}
        Str (r,buf); {raqamni satrga aylantirish}
        s:=buf+s;
    until (n<0);
    Str (n,buf);
    writeln(s);
    readln;
end.
```



{O'nlik sonni o'n oltilikka aylantirish}

uses Crt;

var

n:integer; {Boshlang'ich son}

r:integer; {Sonni sanoq tizimining asosiga bo'lgandagi qoldiq}

s:string [16]; {Sonni berilgan sanoq tizimida ifodalash}

buf: string [1];

begin

Clr Scr

write ('Butun sonni kiriting →');

readln (n);

s:= '';

{Berilgan sonni sanoq son tizimi asosiga (16), bo'lishdan chiqqan qoldiq tizim asosidan katta bo'lgunga qadar, bo'lamiz. Har bir qadamdagi bo'lish qoldig'i — navbatdagi son}

write (n, 'O'nlik songa o'n oltilik son mos keladi');

repeat

r:=n mod 16; {navbatdagi raqam};

n:= n div 16; {bo'lishning butun qismi};

if r<0

then buf:= chr (r+48) {chr(48)= '0', chr (49) = '1' va hokazo}

else buf:= chr (r+55) {chr(65)= 'A', chr (66) = 'B' va hokazo}

s:=buf + s;

until (n<16)

if n <> 0 then begin

if n<0

then buf:= chr (n+48)

else buf:= chr (n+55);

s:=buf + s;

end;

writeln (s);

readln;

end.

{Dastur klaviaturadan kiritilgan arifmetik ifoda qiymatini hisoblaydi}

```
uses Crt;
var
  s: string [80];      {satr}
  l: integer;          {satr uzunligi}
  k: integer;          {ifoda qiymati}
  n: integer;          {navbatdagi son}
  i: integer;          {satrdagi son raqami}
  err: integer;        {belgini songa aylantirishdagi xatolik kodi}
begin
  ClrScr;
  writeln ('Arifmetik ifodani kiriting,');
  writeln ('masalan, 4+5-3-5+2 va <Enter>ni bosing');
  write ('→ ');
  readln (s);
  val (s[1],k,err);
  i:=3;
  repeat
  val (s[i],n,lrr); {navbatdagi bir xonali sonni hosil qiling}
  if s [i-1] = ' +'
      then k:=k+n
      else k:= k-n;
      i:=i+2;
  until i > Length(s);
  writeln ('Kiritilgan ifoda qiymati: ',k);
  writeln ('Dastur ishini yakunlash uchun <Enter>ni bosing.');
```

end.

3.1.9. Massivlarga oid dasturlar

{Massivning nolmas elementlar miqdorini hisoblash}

```
const
  SIZE=5; {massiv o'lchami}
var
  a: array [1 .. SIZE] of integer;
  n: integer; {nolmas elementlar miqdori}
```

```

    i: integer; {indeks}
begin
    writeln ('Butun sonlar massivini kiritish. ');
    write ('Har bir son kiritilgandan so'ng');
    writeln ('<Enter>ni bosing');
    n:=0;
    for i:= 1 to SIZE do
        begin
            write ('a [' ,i,'] → ');
            readln (a[i]);
            if a[i] <> 0 then n:= n+1;
        end;
    writeln ('Massivda', n ta,'nolmas element bor. ');
    readln;
end.

```



124-masala

{Massivning minimal elementini topish}

```

const
    n=5; {massiv o'lchami}
var
    a: array [1 . . n] of integer;
    min: integer; {massiv minimal elementi}
    i : integer;
begin
    writeln ('Massivning minimal elementini topish. ');
    writeln ('Bir satrdan massivning butun sonlar elementlarini')
    writeln (n, ' kiriting va <Enter>ni bosing. ');
    write ('→');
    for i:=1 to n - 1 do
        read (a[i]);
    readln (a[n]);    {dasturning oxiridagi readln ko'rsatma
                       to'g'ri ishlashi uchun}
    min:= a[i];      {faraz qilamiz, birinchi element minimal}
                    {massivning qolgan elementlarini minimal bilan taqqoslaymiz}
    for i:=2 to n do
        if a[i] < min then min:= a[i];
    writeln ('Massivning minimal elementi:', min);
    readln;
end.

```

125-masala

{Havoning haftalik o'rtacha haroratini hisoblash}

const

{hafta kunlari – satrli o'zgarmlar massividir}

*day: array [1..7] of string [11] = ('Dushanba', 'Seshanba',
'Chorshanba', 'Payshanba',
'Juma', 'Shanba', 'Yakshanba');*

var

temper:array [1..7] of real; {harorat}

sum:real; {haftalik harorat yig'indisi}

o'rta: real; {haftalik o'rtacha harorat}

i:integer;

begin

writeln ('Haftalik havo haroratlarini bering.');

for i:=1 to 7 do

begin

write (day [i], '→');

readln (temper[i]);

end;

{Haftalik o'rtacha haroratni hisoblash}

sum:=0;

for i:=1 to 7 do

sum:=sum + temper [i];

o'rta:=sum/7;

writeln ('O'rtacha haftalik harorat:', o'rta:6:2);

readln;

end.

126-masala

{Massivda elementlarni ko'rib chiqish usuli bilan izlash}

var

massiv:array [1..10] of integer; {butun massiv}

namuna:integer; {tekstda izlash uchun namuna}

topildi:boolean; {namuna bilan moslik belgisi}

i:integer;

begin

{10 ta butun sonni kiritish}

writeln ('Massivda izlash.');

write ('Bir satrdan 10 ta sonni bo'shliq orqali kiriting');

```

writeln ('va <Enter>ni bosing');
write ('→');
for i:=1 to 10 do read (massiv[i]);
{sonlar massivga kiritildi}
write ('Izlash uchun namuna kiriting (butun son)→');
readln (namuna);
{oddiy ko'rib chiqish bilan izlash}
topildi:=FALSE; {moslik yo'q}
i:=1; {massivning birinchi elementidan boshlab tekshiramiz}
repeat
  if massiv[i] = namuna
    then topildi:=TRUE {namuna bilan mos tushdi}
    else i:=i+1; {navbatdagi elementga o'tish}
    until (topildi) or (i>10); {agar namuna bilan moslik
                                bo'lsa yoki massivning oxirgi
                                elementi tekshirilsa, ish tugaydi}

if topildi
  then writeln ('Namuna bilan mos tushgan element raqami', i:3,',')
  else writeln ('Namuna bilan moslik yo'q. ');
readln;
end.

```



132-masala

{Massiv elementlari o'sib boruvchi ketma-ketlik hosil qilishini tekshirish}

```

const
  n =5;
var
  a:array [1. . n] of integer; {massiv}
  k:integer; {indeks}
  ok:boolean; {ok = True — kamaymaydigan ketma-ketlik}
begin
  writeln ('Element massivlari o'sib boruvchi ketma-ketlik');
  writeln ('hosil qilishini tekshirish');
  write ('Bir satrdan butun sonlarni', n, ' kiriting');
  writeln ('va <Enter>ni bosing');
  for k:=1 to n - 1 do
    read (a[k]);
  readln (a[n]);
  k:=1;

```

```

ok:=TRUE;
repeat
  if a[k] < a[k+1]
    then k:=k+1
    else ok:=FALSE;
until (ok =FALSE) or (k = n);
write ('Kiritilgan ketma-ketlik');
if not ok
  then write ('o'sib bormaydigan ketma-ketlik');
writeln ('o'sib boruvchi ketma-ketlik.');
```

readln;

end.

133-masala

{Klaviaturadan kiritilgan son massivda necha marta uchrashishini tekshirish}

```

const
  m=5;
var
  a: array [1.. m] of integer; {massiv}
  nam: integer;                {namuna}
  n: integer; {qiymati namunaga teng massivdagi elementlar
  miqdori}
  i: integer; {indeks}
begin
  write ('Massivni kiriting — ');
  writeln (m,'bir satrdagi butun sonlar.');
```

write ('→');

for i:=1 to m - 1 do

read(a[i]);

readln(a[m]);

write ('Taqqoslash uchun namuna kiriting →');

readln (nam);

n:=0;

{qayta ishlash}

for i:=1 to m do

if a[i] = nam then n:=n+1;

if n <> 0

then writeln (nam, 'soni massivda', n, 'marta uchraydi')

*else writeln ('Massivning birona elementi
namunaga teng emas.');*

readln;

end.



139-masala

{Massivni to'g'ridan-to'g'ri tanlash usuli bilan o'sib borishini saralash}

const

n =5; {Massiv o'lchami}

var

a:array [1 .. n] of integer;

i:integer; {minimal elementni izlash boshlangan element raqami}

min:integer; {massivning i dan yuqori chegarasigacha sohadagi element raqami}

j: integer; {minimal element bilan taqqoslanadigan element raqami}

buf:integer; {massiv elementlarini almashtirishda ishlatiladigan bufer}

k: integer; {kiritish, chiqarish uchun indeks}

begin

writeln ('Massivni saralash.');

write ('Bir satrdan', n, 'ta butun sonlarni kiriting');

writeln ('va <Enter>ni bosing');

write ('→');

for k:=1 to n-1

do read (a[k]);

readln (a[n]);

writeln ('Sarlash');

for i:=1 to n-1 do

begin

{massivning a[i] dan a[n] gacha qismida minimal elementni izlash}

min:=i;

for j:=i + 1 to n do

if a[j]<a[min] then min: = j;

{a [min] va a[i] larning o'rnini almashtirish}

buf:= a[i];

a[i]:=a[min];

a[min]:=buf;

```

    {saralash sikli tugadi}
    {massivning oraliq holatini chiqarish}
    for k:=1 to n do write (a[k], ' ');
    writeln;
end;
writeln;
{saralangan massivni chiqarish}
writeln ('Massiv saralangan. ');
for k:=1 to n do
    write (a[k], ' ');
readln;
end.

```



140-masala

{«Ko'piklar» usuli bilan massiv elementlarini o'sib borish tarzida tartibga solish}

```

const
    n=5;
var
    a:array [1.. n] of integer;
    i:integer; {Sikl hisobchisi}
    k:integer; {Massiv elementining joriy indeksi}
    buf:integer;
begin
    writeln ('«Ko'pik» usuli bilan massivni saralash. ');
    write ('Bir satrdan', n, 'ta butun sonni kiriting')
    writeln ('va <Enter>ni bosing');
    for k:=1 to n-1 do
        read(a[k]);
    readln (a[n]);
    writeln ('Saralash. ');
    for i:=1 to n-1 do
        begin
            for k:=1 to n-1 do
                begin
                    if a[k] > a[k+1] then

```

```

        a[k+1]:=buf;
    end;
end;
{Navbatdagi saralashdan keyin massiv holatini chop
etish}
for k:=1 to n do write (a[k], ' ');
writeln;
end;
writeln;
writeln ('Massiv saralandi. ');
for k:=1 to n do write(a[k], ' ');
readln;
end.

```



141-masala

{Ikkita tartibga solingan massivlarni birlashtirish}

```

const
    n = 5;    {boshlang'ich massivlar o'lchami}
var
    a,b: array [1.. n] of integer; {boshlang'ich massivlar}
    c; array [1..2*n] of integer; {natija — massiv}
    k,l,m: integer;                {a,b va c massiv indekslari}
begin
    writeln ('Ikkita tartibga solingan massivlarni o'sib borishiga
            ko'ra birlashtirish. ');
    writeln ('Bitta satrdan birinchi element massivlarini kiriting');
    write ('(, n, ta butun sonni)→');
    for k:= 1 to n-1 do read (a[k]);
    readln (a [n] );
    writeln ('Bitta satrdan ikkinchi massiv elementlarini kiritish');
    write ('(, n, ta butun sonni)→');
    for l:=1 to n-1 do read (b[l]);
    readln (b[n]);
    k:=1; l:=1; m:= 1;
    repeat
        if a[k] < b[l]
            then begin c[m] :=a[k];
                    m:=m+1;
                    k:=k+1;
                end
        else if a[k] > b[l]

```

```

        then begin c[m]:=b[l];
                m:=m+1;
                l:=l+1;
        end
    else begin c[m] :=a[k]
              c[m+1]:=b[l];
              k:=k+1;
              l:=l+1;
              m:=m+2;
        end;
until (k>n) or (l>n); {ikkita boshlang'ich massivlardan bittasi
                    C massivga to'la yozildi}
while k <= n do {A ning C ga yozilmagan elementlari bor}
begin
    c[m] :=a[k] ;
    k:=k+1;
    m:=m+1;
end;
while l <= n do {B ning C ga yozilmagan elementlari bor}
begin
    c[m]:=b[l];
    l:=l+1;
    m:=m+l;
end;
writeln ('Natija — massiv');
for i:=1 to 2*n do
    write [c[i], ' ');
writeln;
writeln ('Dastur ishini tugatish uchun <Enter>ni bosing.')}
readln;
end.

```



142-masala

{Tartibga solingan massivda binar izlash}

```

label 5;
const m=10;
var
    a:array [1 .. 10] of integer; {butur sonlar massivi}
    nam: integer;                {izlash uchun namuna}
    ok: boolean;                 {TRUE — massiv tartibga
                                solingan}

```

```

o'rtta,yuqori,quyi:integer; {massivning o'rtta, yuqori va
                               quyi elementlari raqami}
found:boolean;                {namuna bilan mos kelish belgisi}
n:integer;                    {namuna bilan moslik hisobchisi}
i:integer;

begin
  {massivni kiritish}
  writeln ('*** Tartibga solingan massivda binar izlash ***');
  write ('Massivni kiriting (bir satrdan', m,' ta butun sonni');
        writeln ('va <Enter>ni bosing');
  write ('→');
  for i:=1 to m-1 do read(a[i]);
  readln (a[m]);
    {Massiv tartibga solinganini tekshirish}
  i:=1;
  ok:=TRUE; {massiv tartibga solingan bo'lsin}
  while (i < HB) and ok
    do if a[i] <= a [i+1]
       then i:=i+1
       else ok:=FALSE;
  if not ok then
  begin
    writeln ('Kiritilgan massiv o'sib borishi bo'yicha tartibga
              solinmagan');
    goto 5;
  end;
  write ('Izlash uchun namuna (butun son) kiriting →'),
  readln (nam);
  {binar izlash}
  yuqori:=1;
  quyi:=HB;
  found:=FALSE;
  n:=0;
  repeat
    o'rtta:=(quyi-yuqori) div 2 + yuqori; {massivni teng ikkiga
      bo'lamiz}
    n:= n+1;
    if a [o'rtta] = nam
      then found:=TRUE
      else

```

```

        {izlanayotgan element qaysi qismda, yuqori yoki
        pastda bo'lishini aniqlash}
        if nam < a[o'rta]
            then quyi:= o'rta - 1
            else yuqori:= o'rta + 1;
        until (yuqori > quyi) or found;
        if found
            then write (o'rta, '-raqamli element bilan mos tushish', n,
            ' ta taqqoslash olib borildi.')
            else writeln ('Massivda namuna topilmadi.');
```

5: readln;
end.



143-masala

{Talabalarning bo'yi xususidagi ma'lumot tahlili}

```

const
    max=30; {sinfdagi o'quvchilarning eng ko'p soni}
var
    bo'y: real; {o'quvchi bo'yi}
    guruh: array [1.. max] of real; {o'quvchilar bo'yi}
    n: integer; {ma'lumotlari kiritilgan o'quvchilar soni}
    o'rta: real; {o'rta bo'y}
    m: integer; {bo'yi o'rtadan yuqori o'quvchilar soni}
    sum: real; {jami bo'y}
    i: integer;
begin
    writeln ('*** O'quvchilar bo'yining tahlili ***');
    writeln ('Bo'y uzunligi (sm)ni kiriting va <Enter>ni bosing');
    writeln ('Ishni tugatish uchun 0 ni kiriting va <Enter>ni
    bosing');
```

n:=0;

```

repeat
    write ('→');
    readln (bo'y);
    if bo'y <> 0 then
        begin
            n:=n+1;
            guruh [n]:= bo'y;
            sum:=sum+ bo'y;
        end;
end;
```

```

until (n > max) or (bo'y = 0);
if n > 0 then
  begin
o'rta: = sum/n;
m:=0;
    {har bir bo'yni o'rtachasi bilan taqqoslaymiz}
for i:=1 to n do
  if guruh [i] > o'rta then m:=m+1;
  writeln ('O'rtacha bo'y:', o'rta;6:l,'sm');
  writeln (m,'-ta o'quvchining bo'yi o'rtachadan baland');
end
    else writeln ('Qayta ishlash uchun ma'lumotlar
    yo'q. ');
  readln;
end.

```



144-masala

{Ustunlar bo'yicha massiv elementlarining yig'indisini hisoblash}

```

const
  m =3; {satrlar soni}
  n =5; {ustunlar soni}
var
  a: array [1.. m, 1.. n] of integer; {massiv}
  s: array [1.. n] of integer; {elementlar yig'indisi}
  i,j: integer;
begin
  writeln ('Massivni kiriting. ');
  writeln ('Har bir,' n, 'satrdagi butun son elementlarini',
    'kiritgandan keyin <Enter>ni bosing');
  for i:=1 to m do {m satr}
    begin
      write('→');
      for j:=1 to n -1 do
        read (a[i,j]);
      readln (a[i, n]);
    end;
  writeln ('Kiritilgan massiv');
  for i:=1 to m do
    begin
      for j:=1 to n -1 do

```

```

        write (a[i,j]:4);
        writeln (a[i, n]:4);
    end;
    {Qayta ishlash}
    for j:=1 to n do {har bir ustun uchun bitta ustun elementlarini}
        {qo'shamiz}
        for i:=1 to m do s[j]:=s[j] + a [i,j];
        writeln ('-----');
        for i:=1 to n do write (s[i]:4);
        writeln;
        readln;
    end.

```



145-masala

{Ikkinchi tartibli matritsa aniqlovchisini hisoblash}

```

var
    a: array [1..2,1..2] of real;
    det: real;           {aniqlovchi (determinant)}
    i,j: integer;       {massiv indekslari}
begin
    writeln ('Ikkinchi tartibli matritsani kiriting. ');
    writeln ('Satr elementlari kiritilgach, <Enter>ni bosing');
    for i:=1 to 2 do
        begin
            write ('→');
            read (a[i,1]);
            readln (a[i,2]);
        end;
    det:=a[1,1]*a[2,2] - a[1,2] * a[2,1];
    writeln ('Matritsa aniqlovchisi');
    for i:=1 to 2 do
        begin
            for j:=1 to 2 do
                write (a[i,j]:6:2);
            writeln;
        end;
    writeln (det:6:2,'ga teng');
    readln;
end.

```




{Matritsa «afsunkor» kvadrat bo'lishini tekshiring}

const

MAX = 5; {matritsaning maksimal o'lchami}

var

a: array [1..MAX, 1..MAX] of integer; {matritsa}

n: integer; {tekshirilayotgan matritsa o'lchami}

ok:boolean; {TRUE — matritsa afsunkor kvadrat bo'ladi}

i,j: integer; {massiv indeksleri}

sum: integer; {matritsa bosh diagonali elementlari yig'indisi}

s: integer; {joriy satr, ustun yoki matritsa ikkinchi diagonali elementlarining yig'indisi}

begin

write ('Matritsa elementlarini kiriting (3..4) ', 'va
<Enter>ni bosing→');

readln(n);

writeln ('Matritsa satrlarini kiriting');

writeln (n, 'satrning butun sonlarini kiritgandan keyin
<Enter>ni bosing');

for i:=1 to n do

begin

write ('→')

for j:=1 to n-1 do read (a[i, j]);

readln (a[i, n]);

end;

ok:=TRUE; {matritsa «afsunkor» kvadrat bo'lsin}

sum:=0;

{bosh diagonal elementlari yig'indisini
hisoblaymiz}

for i:=1 to n do

sum:=sum +a[i, i];

{satrlar bo'yicha yig'indilarni hisoblaymiz}

i:=1;

repeat

s:=0; {joriy satr elementlari yig'indisi}

for j:=1 to n do s:=s+a [i, j];

i:=i+1;

if s <> sum then ok:=FALSE;

until (not ok) or (i > n);

if ok then

{bu yerda har bir satr elementlari yig'indisi bosh diagonal elementlari yig'indisiga teng}

begin

{ustunlar bo'yicha yig'indilarni hisoblaymiz}

j:=1;

repeat

s:=0; {joriy ustun elementlari yig'indisi}

for i:=1 to n do s:=s+a[i,j];

j:=j+1;

if s <> sum then ok:=FALSE;

until (not ok) or (j > n);

if ok then

{bu yerda har bir satr elementlari yig'indisi har bir ustun elementlari yig'indisi va bosh diagonal elementlari yig'indisiga teng}

begin

{ikkinchi bosh diagonal elementlari yig'indisini hisoblaymiz}

s:=0;

j:=n;

for i:=1 to n do

begin

s:= s+a[i, j];

j:=j - 1

end;

if temp <> sum then ok:=FALSE;

end;

end;

write ('Kiritilgan matritsa');

if not ok

then write ('«afsunkor» kvadrat bo'lmaydi');

writeln ('«afsunkor» kvadrat bo'ladi.');

readln;

end.



147-masala

{Universiada o'yinlari natijalarini yakunlash}

const

n=14; {ishtirokchi viloyatlar soni}

viloyat: array [1..n] of string [9]=('Andijon', 'Buxoro', 'Jizzax', 'Navoiy', 'Namangan', 'Samarqand', 'Surxondaryo'

'Sirdaryo', 'Toshkent', 'Farg'ona', 'Xorazm',
'Qoraqal. Resp.' 'Qashqadaryo', 'Toshkent sh');

var

{natijalar jadvali}

natija: array [1.. $n+1$, 1..5] of integer;

{ $n+1$ — satr jadvalni saralashda bufer sifatida ishlatiladi}

i, j: integer;

max: integer; {ochkolar soni maksimal darajada
saralashda ishlatiladigan jadval satri raqami}

buf: string [9];

begin

writeln ('Universiada o'yinlari natijasi');

writeln ('Bir satrdan oltin, kumush va bronza medallari sonini
kiriting.');

{boshlang'ich ma'lumotlarni kiritish}

for $i:=1$ to n do

begin

write (viloyat [i], ' → ');

read (natija [i , 1], natija [i , 2]); {oltin va kumush medallar
soni}

readln (natija [i , 3]); {bronza medallar soni}

end;

{umumiy medallar va ochkolar sonini hisoblaymiz}

for $i:=1$ to n do

begin

natija [i , 4]:= natija [i , 1]+ natija [i , 2]+ natija [i , 3];

na ija [i , 5]:= natija [i , 1]*7+ natija [i , 2]*6+ natija [i , 3]*5;

end;

{oddiy tanlash usuli bilan ochkolar soniga mos ravishda
massivlarni saralash}

for $i:=1$ to $n-1$ do

begin

{jadval qismidan i -satrdan boshlab, natija [j , 5] element
maksimal bo'lgan j -satrni topish}

max:= i ; {bu i -raqamli satr bo'lsin}

for $j:=i+1$ to n do

if natija [j , 5] > natija [max , 5] then *max*:= j ;

{*max* raqamli satrni i raqamli satr bilan almashtirish.}

{Bufer sifatida, jadvalning ishlatilmaydigan oxirgi satri
foydalaniladi.}

```

    buf:= viloyat [i];
    viloyat [i]:= viloyat [max];
    viloyat [max]:=buf;
    for j:=1 to 5 do
        begin
            natija [n+1,j]:= natija [i,j];
        end;
    for j:=1 to 5 do
        begin
            natija [i,j]:= natija [max,j];
        end;
        for j:=1 to 5 do
            begin
                natija [max,j]i:= natija [n+1,j];
            end;
        end;
        (bu yerda, jadval tartibga solingan)
    writeln;
    writeln ('Talabalar Universiadasi yakunlari:');
    writeln ('Viloyat':12 , 'Oltin':8, 'Kumush':8, 'Bronza':8, 'Jami':8,
        'Ochkolar':8);
    for i:=1 to n do
        begin
            write (i:2, viloyat [i]:10);
            for j:=1 to 5 do
                write (natija [i,j]:8);
            writeln;
        end;
    readln;
end.

```



148-masala

{«Sonni top» o'yini — murakkablashtirilgan}

const

n=3 {murakkablik darajasi — sondagi raqamlar miqdori}

var

o'yinchi: array [1..n] of char; (o'yinchi kombinatsiyasi)

comp: array [1..n] of char; (kompyuter kombinatsiyasi)

a: array [1..n] of boolean; {agar kompyuterning i — raqamli o'yinchi raqamlaridan biron tasi bilan mos tushsa, a [i] = TRUE}

```

top:integer;      {son topildi}
o'rin:integer;   {ulardan o'z o'rinda}
i,j: integer;    {massiv indeksleri}
begin
  writeln        ('Kompyuter uch xonali sonni o'yladi.',
                 'Siz uni topishingiz kerak,');
  writeln        ('Navbatdagi son kiritilgandan keyin Sizga nechta
                 raqam');
  writeln        ('topilganligi ulardan nechtasi o'z o'rnida ekanligi
                 aytiladi');
  writeln        ('Sonlar kiritilgach <Enter>ni bosing. ');
                 {Kompyuter o'z sonini «o'ylaydi»}
  randomize;
  for i:=1 to n do
    comp [i]:= chr (random (10) +49); {48 - '0' belgi kodi}
    write ('Kompyuter o'yladi:'); for i:=1 to n do
write (comp [i]); writeln;
  repeat
    write ('Sizning variantingiz → ');
    {o'yinchi variantini olish}
    for i:=1 to n-1 do
      read (o'yinchi [i]);
    readln (o'yinchi [n]);
    for i:=1 to n do a[i]:=FALSE;
    {a[i] = TRUE, agar kompyuter sonining i-chi raqami
    o'yinchi soni raqamlaridan bittasi bilan
    mos kelsa, nechta raqam topilganligi
    tekshiriladi}

    top:=0;
    for i:=1 to n do {o'yinchining har bir raqami tekshiriladi}
      for j:=1 to n do
        begin
          if (o'yinchi [i] = comp[j]) and not a[j] then
            begin
              top:= top+1;
              a[j]:=TRUE; {kompyuterning bu ra-
              qamini o'yinchining tekshiril-
              magan boshqa qolgan raqamlari
              bilan taqqoslash man etiladi}
            end;
        end;
      end;
    end;
end;

```

```

    {ulardan nechitasi o'z o'rnida ekanligi tekshiriladi}
    o'rin:=0;
  for i:=1 to n do
    if o'yinchi [i] = comp[i] then o'rin:= o'rin+1;
    writeln ('Topildi:', top, '. O'z o'rnida:', o'rin);
  until (top=n) and (o'rin=n);
  writeln ('***SONNI TOPDINGIZ!***');
  write ('Ishni tugatish uchun <Enter>ni bosing. ');
  readln;
end.

```



149-masala

{Telegraf — Morze alifbosi yordamida axborotni uzatish}

(Izoh: to'xtatishlar kattaligini o'stirish mumkin)

uses Crt;

const

```

  morse:   array [128 .. 159] of string [4] =
    ( ' .', ' -...'; ' ---', ' ---', {A,B,B,Г}
      ' -..', ' ...', ' ---', ' ---', {Д,Е,Ж,З}
      ' ..', ' ---', ' ---', ' ---', {И,Й,К,Л}
      ' ---', ' ---', ' ---', ' ---', {М,Н,О,П}
      ' -.', ' ...', ' ...', ' ...', {Р,С,Т,У}
      ' ---', ' ...', ' ---', ' ---', {Ф,Х,Ц,Ч}
      ' ---', ' ---', ' ---', ' ---', {Ш,Щ,Ъ,Ы}
      ' ---', ' ---', ' ---', ' ---', {Ь,Э,Ю,Я}

```

{uzatish parametrlari}

TONE =100; {signal chastotasi (gs)}

L1=50; {«nuqta» uzunligi (ms)}

L2=100; {«tire» uzunligi (ms)}

L3=50; {nuqtalar va bitta harf tiresi o'rtasida sukut pauza (ms)}

L4=100; {harflar o'rtasidagi sukut-pauza (ms)}

L5=150; {so'zlar o'rtasidagi sukut-pauza (ms)}

var

mes: string; {axborot}

sim: string [4]; {Morzeda ramzlarni kodlashtirish nuqta va tirelar ketma-ketligidan iborat}

```

belgi: string [1]; {«uzatiladigan» belgi — tire yoki nuqta}
i,j: integer;      {ramz va belgi raqami}
begin
  ClrScr;
  writeln ('*** Telegraf ***');
  writeln ('Uzatilishi kerak bo'lgan axborotni kiriting');
  writeln ('(faqat bosh kirill harflaridan foydalaning)');
  write ('→');
  readln (mes);
  for i:=1 to Length(mes) do
    begin
      if (mes[i] >= 'A') and (mes[i] <='Я') then
        begin
          {axborotning navbatdagi harf kodini aniqlash (f-я
          Ord) va kodlar jadvalidan massivning tegishli
          elementi — nuqta va tirelar ketma-ketligini hosil
          qilish}
          sim:=morse [ord (mes[i])];
          j:=1;
          repeat
            if (sim[j] = '-') or (sim[j] = '.') then
              begin
                write (sim[j]);
                sound (1000);
                case sim [j] of
                  '.': Delay (50);
                  '-': Delay (100);
                end;
                NoSound;
                Delay (50);
              end;
            j:=j+1;
          until ((sim [j] = ' ') or (j>4));
          Delay (100); {harflar o'rtasida sukut-pauza}
        end
      else
        if mes [i] = ' ' then {so'zlar o'rtasida bo'shliq}
          begin
            write (' '); axborot
            so'zlari o'rtasida bo'shliq}
            Delay (150);
          end;
    end;
end

```

```
writeln;
writeln ('Axborot uzatildi!');
writeln ('Dastur bilan ishni tugatish uchun <Enter>ni
        bosing');
readln;
end.
```



3.1.10. Funksiyalar ishtirokidagi dasturlar



150-masala

{max funksiya ikki sondan biriga katta qiymatni beradi}

```
function max (a,b: integer): integer;
begin
    if a > b
        then max:=a
        else max:=b;
end.
```



151-masala

{Quyidagi funksiya sonlarni taqqoslash natijasini munosabat belgilari ko'rinishida beradi}

```
function Compare (a,b: real): char;
begin
    if a > b then Compare:='>'
    else
        if a < b then Compare:='<'
        else Compare:='=';
end;
var
    x1,x2: real;      {taqqoslanadigan sonlar}
    nat: char;       {taqqoslash natijasi}
begin
    writeln ('Ikki sonni kiriting va <Enter>ni bosing');
    write ('→');
    readln (x1,x2);
    nat:= Compare (x1, x2);  {dasturchi funksiyasini chaqirish}
    writeln (x1:6:2, nat, x2:6:2);
    readln;
end.
```




152-masala

*{Elektr zanjiri qarshiligini hisoblash}**function Qar (r1,r2: real; t: integer): real;**{r1,r2 — qarshilik qiymatlari, t — ulanish turi:**1 — ketma-ket;**2 — parallel.**Agar ulanish turi xato ko'rsatilgan bo'lsa, 1 ni beradi}**begin**if t=1 then Qar:= r1+r2;**if t=2 then Qar:= r1*r2/(r1+r2)**else Qar:=1; {ulanish turi xato ko'rsatilgan}**end.*

153-masala

*{Logarifm xossalaridan foydalanib, son darajasini hisoblash}**function InDar (a,b:real):real;**begin**{A ning B darajasi C ga teng.**Tenglikning ikkala tomoni logarifmlanib, quyidagi hosil qilanadi:* *$B * \ln(A) = \ln(C)$* *C qiymatni topish kerak bo'lgani uchun E ning $b * \ln(A)$ darajasi hisoblanadi. Bu ifodaning qiymati C ga teng, shuni hisoblash talab qilingan edi.}**InDar:=exp (b*ln(a));**end;**var**a: real; {son}**b: real; {daraja}**c: real; {darajadagi son}**begin**writeln ('Sonni va daraja ko'rsatkichini kiriting');**readln (a,b);**c:= InDar (a,b);**writeln (a:6:3, ' darajasi ',b:6:3, ' = ',c:6:3);**readln;**end.*



154-masala

{Omonatga ko'ra daromadni hisoblash}

```
function Daromad (sum: real;      {omonat jami}
                  omonat: real; {yillik omonat foizi}
                  muddat: integer {omonat muddati (kun)}):
    real;

begin
    Daromad:= sum*(omonat/100/365)* muddat; {365
        yildagi kunlar soni}
end.
```



155-masala

{Belgi unli harf bo'lishini tekshirish}

```
Function Unli (sim:char): boolean;
const
    {unli harflar}
    ListofUnli: string = 'AaEeliOoUuEeYuyuYaya O'o';
var
    p: byte; {tekshirilayotgan belgining unliharf bo'lganligini}
begin
    p:=Ord(sim);
    if p <> 0 {belgi ro'yxatda topildi}
    then Unli:=True
    else Unli:=False;
end.
```



156-masala

{Satrdan boshlang'ich bo'shliqlarni olib tashlash}

```
function LTrim (s: string): string;
begin
    while (Ord(s[1]) = 32) and (length (s) > 0) do
        delete (s, 1, 1);
    LTrim:= s;
end;
{LTrim funksiya ishini tekshirish}
var
    s:string [80];      {s: string}
begin
    writeln ('Satrdan boshlang'ich bo'shliqlarni olib tashlaydi.');
```

```

write ('Satrni kiriting →');
readln (s);
write ('Boshlang'ich bo'shliqlarsiz satr:', LTrim (s));
readln;

```

end.



157-masala

{Bosma harflarni yozma harflarga aylantirish}

{UpCase standart protsedurasi o'rnini bosadi}

function UpCase (satr:string): string;

var

i:integer;

begin

for i:=0 to Length (satr) do {belgilar noldan boshlab
raqamlanadi!}

case satr [i] of

{lotin harflari}

'a'..'z': UpCase[i]:=chr(ord(satr [i])-32);

{rus harflari}

'a'..'n': UpCase[i]:=chr(ord(satr [i])-32);

'p'..'я': UpCase[i]:=chr(ord(satr [i])-80);

else {qolgan belgilar almashtirilmaydi}

UpCase[i]:=st[i];

end;

{UpCase funksiyasini ishlatishga misol}

var

satr: string;

begin

writeln ('Matnni kiriting va <Enter>ni bosing');

write ('→');

readln (satr);

writeln (UpCase (satr));

readln;

end.



158-masala

{Kvadrat tenglamani yechish}

function KvadrT (a,b,c: real; var x1,x2: real): integer;

{a,b,c — tenglama koeffitsientlari}

```

{x1,x2 – tenglama ildizlari}
{funksiya qiymati — ildizlar soni yoki -1, agar
boshlang'ich qiymatlar xato bo'lsa,}
var
  d: real; {diskriminant}
begin
  if a = 0 then KvadrT:= -1
  else
    begin
      d:=b*b - 4*a*c;
      if d < 0 then
        KvadrT:= 0 {tenglama
yechimga ega emas}
      else
        begin
          if d > 0
            then KvadrT:=2
            {ikkita har xil ildizlar}
          else KvadrT:=1;
          {ildizlar bir xil}
          x1:= (-b+Sqrt(d))/(2*a);
          x2:= (-b-Sqrt(d))/(2*a);
          end;
        end;
    end;
end,
{asosiy dastur}
var
  a,b,c: real; {tenglama koeffitsientlari}
  x1,x2: real; {tenglama ildizlari}
begin
  writeln ('Kvadrat tenglamani yechish');
  writeln ('Bitta satrda koeffitsientlarni kiriting va <Enter>ni
bosing');
  write ('→');
  readln (a,b,c);
  case KvadrT (a,b,c,x1,x2) of
    -1: writeln ('Boshlang'ich qiymatlar xatosi. ');
    0: writeln ('Tenglama yechimga ega emas. ');
    1: writeln ('x=',x1:6:2, 'Ildizlar bir xil. ');
    2: writeln ('x1=', x1:6:2, 'x2=', x2: 6:2) ;
  end;
  readln;
end.

```

uses Crt;

var

a: integer; {foydalanuvchi kiritgan son}
 {Funksiya *GetInt* bitta yoki ikkita raqamdan iborat
 butun musbat sonni kiritish uchun ishlatiladi.
 Kiritish vaqtida tahrirlash uchun <Backspace>ni
 ishlatish mumkin.
 <Enter> bosilganda funksiya kiritilgan sonni
 qaytaradi.}

function *GetInt*: integer;

const

K_BACK=8; {<Backspace> klavishi kodi}
 K_ENTER=13; {<Enter> klavishi kodi}

var

ch:char; {belgi}
 dec:byte; {belgi kodi}
 buf:array [1..2] of char; {kiritilgan raqamlar}
 n: integer; {kiritilgan raqamlar soni}
 x,y: integer; {kursor o'rni}

begin

buf [1]:=' '; buf[2]:=' ';
 n:= 0;

repeat

ch:=Readkey;
 dec:=ord(ch);
 if (ch>= '0') and (ch<='9') and (n<2) then
 begin

write (ch);
 n:=n+1;
 buf [n] r=ch;

end

else

if (dec=K_BACK) and (n>0) then

begin
 n:= n-1;
 x:= WhereX;
 y:= WhereY;
 GotoXY (x-1,y);
 write (' ');
 GotoXY (x-1, y) ;

end;

```

until (n>0) and (dec=K_ENTER);
{kiritilgan satrni songa aylantiramiz}
if n = 2
  then GetInt:= (ord (buf [1]) - 48) *10 + ord (buf [2] ) - 48
  else GetInt:= ord (buf[1]) - 48;
end;
begin
  ClrScr;
  writeln ('*** GetInt funksiya ishini namoyish etish ***');
  writeln;
  writeln ('GetInt funksiya bitta yoki ikkita raqamdan iborat');
  writeln ('butun musbat sonni kiritishda ishlatiladi. ');
  writeln ('Kiritish paytida tahrirlash uchun <Backspace>
  dan foydalanish mumkin. ');
  writeln ('<Enter> bosilganda funksiya kiritilgan sonni
  qaytaradi. ');

  writeln;
  writeln ('Sonni kiriting va <Enter>ni bosing');
  write ('→');
  a:=GetInt;
  writeln;
  write ('Siz', a, 'sonini kiritdingiz');
  readln;
end.

```



3.1.11. Protsedurali dasturlar



170-masala

{Yulduzchalardan iborat satrni chiqarish}

```

procedure StarLine (len: integer);
var
  i:integer;
begin
  for i:=1 to len do
    write('*');
end.

```



172-masala

{Ramkani chizish}

```

procedure Frame (e:integer; t:integer; w:integer; h:integer);
{e,t –yuqori chap burchak koordinatalari}

```

```

    {w — ramka kengligi, h — ramka balandligi}
var
    x,y:integer; {chiqariladigan belgi koordinatalari}
    i:integer;
    c1,c2,c3,c4,c5,c6:char; {ramkani chizadigan belgilar}
begin
    c1:=chr(218); {yuqori chap burchak belgisi}
        {chr belgi funksiyani chaqirishda raqami
        ko'rsatilgan belgini qaytaradi}
    c2:=chr(196); {gorizontal lineyka belgisi}
    c3:=chr(191); {yuqori o'ng burchak belgisi}
    c4:=chr(179); {vertikal lineyka belgisi}
    c5:=chr(192); {quyi chap burchak belgisi}
    c6:=chr(217); {quyi o'ng burchak belgisi}
    GOTOXY (1,t);
    write (c1);
    for i:=1 to w-2 do {ramka yuqori chegaralari belgilari}
        write (c2);
    write (c3);
    y:=t+1
    x:1+w-1
    for i:=1 to h-2 do {chap va o'ng chegaralar belgilari}
        begin
            GoToXY (1, y);
            write (c4)
            GoToXY (x, y);
            write(c4)
            y:=y+1
        end;
    GoToXY (1,y);
    write (c5)
    for i:=1 to w-2 do {quyi chegara belgilari}
        write (c2);
    write (c6);
end.

```



3.1.12. Fayllarga oid dasturlar



181-masala

{A: diskda fayl yaratish va unga foydalanuvchi tomonidan kiritilgan 5 ta butun sonni yozish}

```

var
  f: text;      {matn fayli}
  n: integer;   {son}
  i: integer;   {son hisobchisi}
begin
  writeln ('Fayl yaratish');
  writeln ('5 ta butun sonni kiriting. ');
  writeln ('Har bir son kiritilgach, <Enter>ni bosing');
  Assign (f, 'a:\numbers.txt');
  Rewrite(f); {qayta yozish rejimida oching}
  for i:=1 to 5 do
    begin
      write ('→');
      readln(n);
      writeln(f,n);
    end;
  close(f); {faylni yoping}
  writeln ('Kiritilgan sonlar 'a:\nnumbers.txt faylida yozilgan');
  readln;
end.

```



182-masala

{a:\numbers.txt fayliga foydalanuvchi kiritgan 5 ta butun sonlarni qo'shimcha yozish}

```

var
  f: text;      {matn fayli}
  n: integer;   {son}
  i: integer;   {sonlar hisobchisi}
begin
  writeln (' a:\numbers.txt fayliga qo'shimcha');
  writeln ('Beshta butun son kiriting. ');
  writeln ('Har bir son kiritilgach, <Enter>ni bosing').
  Assign (f, 'a:\numbers.txt');
  Append(f); {qo'shish rejimida faylni ochish}
  for i:=1 to 5 do
    begin
      write ('→'),
      readln (n);
      writeln (f,n);
    end;
end;

```



```

Close(f);          {faylni yopish}
writeln ('Kiritilgan sonlar a:\numbers.txt faylga qo'shilgan');
readln;
end.

```



183-masala

{a:\numbers.txt faylning tarkibini ekranga chiqarish}

```

var
    f: text; {matn fayli}
    n: integer; {son}
begin
    writeln ('a:\numbers.txt fayl tarkibi');
    writeln ('-----')
    Assign (f, 'a:\numbers.txt');
    Reset (f); {o'qish uchun faylni ochish}
    While not EOF (f) do {hozircha fayl oxiriga yetkazilmagan}
        begin
            readln (f,n);    {fayldan sonni o'qish}
            writeln (n);    {o'qilgan sonni ekranga chiqarish}
        end;
    Close (f);          {faylni yopish}
    writeln ('-----')
    readln;
end.

```



184-masala

{a:\numbers.txt faylidagi sonlarning o'rta arifmetikligini hisoblash}

```

var
    f: text; {matn fayli}
    n: integer; {fayldan o'qilgan son}
    miq: integer; {o'qilgan sonlar miqdori}
    sum: integer; {o'qilgan sonlar jami}
    o'a: real; {o'rta arifmetik}
begin
    writeln ('a:\numbers.txt faylidagi sonlarning,');
    writeln ('o'rta arifmetikligini hisoblash');
    writeln ('Fayldan o'qish. Kuting. ');
    sum:=0;
    miq:=0;

```

```

Assign (f, 'a:\numbers.txt');
Reset (f);           {faylni o'qish uchun ochish}
While not EOF (f) do {hozircha fayl oxiriga yetkazilmagan}
  begin
    readln (f,n);     {fayldan sonni o'qish}
    sum:=sum+n;
    miq:= miq +1;
  end;
Close (f);           {faylni yopish}
o'a:=sum/ miq;
writeln ('O'qilgan sonlar: ', miq);
writeln ('Sonlar yig'indisi: ',sum);
writeln ('O'rta arifmetik: ',o'a:9:2);
readln;
end.

```



185-masala

{Ismi foydalanuvchi tomonidan ko'rsatilgan fayl tarkibini ekranga chiqarish}

```

uses Crt;
var
  f: text;           {matn fayli}
  fname: string [80]; {fayl ismi}
  st: string;        {fayldan o'qilgan satr}
  sts: integer;      {ekranga chiqarilgan satrlar soni}
  key: char;         {foydalanuvchi bosgan klavish}
begin
  ClrScr;           {ekranni tozalash}
  writeln ('Matn faylini ko'zdan kechirish');
  writeln ('Faylning to'la ismini kiriting va <Enter>ni
    bosing');
  write ('→ ');
  readln (fname);
  Assign (f,fname);
  Reset(f);        {faylni o'qish uchun ochish}
  ClrScr;
  sts:=0;
  While not EOF(f) do {hozircha fayl oxiriga yetkazilmagan}
    begin
      readln (f,st);   {fayldan sonni o'qish}

```

```

writeln (st);
sts:=sts+1;
if sts = 23 then {navbatdagi 23 ta satr chiqarildi}
  begin
    writeln;
    write ('Kiritishni davom ettirish uchun
           ixtiyoriy klavishni bosing...');
    key:= Readkey;
    GotoXY (1,WhereY); {kursorni joriy satr boshiga
                       eltish}
    DelLine; {«Davom ettirish uchun...» axborotini
             yo'qotish}
    sts:=0;
  end;
end;
Close (f);      {faylni yoping}
writeln;
write ('Ko'rishni davom ettirish uchun ixtiyoriy klavishni
       bosing...');
key:=Readkey;
end.

```



186-masala

{a:\phone.txt fayliga ism, sharif va telefon raqamini qo'shimcha yozish. Agar fayl diskda bo'lmasa, uni yaratish}

```

label 1;
var
  f: text;           {matn fayli}
  fam: string [15]; {familiya}
  name: string [15]; {ism}
  tel: string [9];  {telefon raqami}
begin
  writeln ('Telefon ma'lumotnomasiga qo'shimcha');
  Assign (f, 'a:\phone.txt');
  {$I}
  Append (f); {avval qo'shish rejimida ochamiz}
  if IOResult <> 0 then
    {fayl phone.txt A: diskda bo'lmaslik ehtimoli bor,}
    {uni yaratamiz}
begin

```

```

Rewrite (f); {qayta yozish rejimida ochish}
if IOResult <> 0 then
  begin
    writeln ('A: diskka murojaat etishda xatolik');
    goto 1;
  end;
end;
{foydalanuvchidan ma'lumotlar olamiz}
write ('Familiya →');
readln (fam);
write ('Ism →');
readln (name);
write ('Telefon →')
readln (tel);
{va ularni faylga yozamiz}
writeln (f,fam);
writeln (f, name);
writeln (f,tel);
close (f);
writeln ('Axborot qo'shildi.');
```

1: writeln ('Ishni tugatish uchun <Enter>ni bosing.');

```

readln;
end.
```



187-masala

{Telefon ma'lumotnomasidan izlash}

```

label 2;
var
  f: text;           {matn fayli}
  nam: string [15]; {izlash uchun familiya}
  n: integer;       {so'rovni qanoatlantiradigan yozuvlar
                    miqdori}
  fam: string [15]; {familiya}
  name: string [15]; {ism}
  tel: string [9];  {telefon raqami}
begin
  writeln ('Telefon ma'lumotnomasidan izlash');
  Assign (f, 'a:\phone.txt');
  {$I-}
  reset (f);        {o'qish uchun faylni ochamiz}
```

```

if IOResult <> 0 then
  {phone.txt fayli A: diskda bo'lmalik ehtimoli bor}
  begin
    writeln ('a:\phone.txt faylidan foydalanib bo'lmaydi);
    goto 2;
  end;
repeat
  write ('Familiya → ');
  readln (nam);
  if Length (nam) <> 0 then {foydalanuvchi satrni kiritdi}
    begin
      reset(f); {faylni boshidan ko'rib chiqamiz}
      n:= 0;
      while not EOF(f) do {faylning hammasini ko'rib
        chiqamiz}
        begin
          {fayldan o'qiymiz}
          readln (f,fam);
          readln (f,name);
          readln (f,tel);
          if fam = nam then {kerakli familiya topildi}
            begin
              writeln (fam,' ',name,' ',tel);
              n:= n+1;
            end;
          end;
        end;
      if n = 0 then writeln (nam, 'to'g'risida axborotlar
        yo'q');
    end;
  until Length (nam) = 0;
2:writeln ('Ishni tugatish uchun <Enter>ni bosing. ');
  readln;
end.

```



188-masala

{Testdan o'tkazishning universal dasturi}

```

uses Crt;
label
  nomer;
var

```

```

fname: string [40]; {testdagi fayl ismi}
f: text; {test fayli}
JamiSav: integer; {test savollari miqdori}
To'g'Jav: integer; {joriy savol uchun to'g'ri javoblar
miqdori}
nJav; integer; {mazmuni bir xil bo'lgan javoblar
soni}
To'g': integer; {to'g'ri javob raqami}
Jav: integer; {foydalanuvchi tanlagan javob
raqami}
sat: string; {test faylidan o'qiladigan satr}
p: integer; {to'g'ri javoblar foizi}
i: integer; {sikllar hisobchisi}
begin
  if ParamCount = 0 then
    begin
      writeln ('Test savollari fayli berilmagan!');
      writeln ('Buyruq satri: test TestFayliIsmi');
      goto nomer;
    end;
  fname: = ParamStr (1); {buyruq satridagi fayl ismi}
  Assign (f,fname);
  {$I-} {bu izoh emas, balki kompilyatorga buyruq}
  Reset(f); {o'qish uchun faylni ochish}
  {$I+}
  if IOResult <> 0 then
    begin
      writeln (fname, 'test fayli topilmadi');
      goto nomer;
    end;
  writeln ('Hozir Sizga test taklif etiladi. ');
  writeln ('Har bir savolga javoblarning bir nechta varianti beriladi);
  writeln ('Siz to'g'ri javob raqamini kiritishingiz va <Enter>
klavishini bosishingiz kerak');
  writeln;
  writeln ('Muvaffaqiyat! Testni boshlashdan oldin <Enter>ni
bosing');
  TextBackGround(Blue);
  ClrScr;
  JamiSav:=0;

```

```

To'g':=0;
while not EOF(f) do
  begin
    {ClrScr;}
    JamiSav:= JamiSav + 1;
    Readln (f,sat);           {fayldan savolni o'qiyimiz}
    TextColor(White);
    writeln(sat);           {Ekranga savolni chiqaramiz}
    readln (f,nJav, To'g');  {mohiyati bir xil bo'lgan
                              javoblar sonini va to'g'ri javob
                              raqamini o'qiyimiz.)}

    TextColor (LightGray);
    for i:=1 to nJav do {o'qiyimiz va mohiyati bir xil javoblarni
                        chiqaramiz}
      begin
        readln (f,sat);
        writeln(i, ' ',sat);
      end;
    writeln;
    write ('Sizning tanlovingiz →');
    readln(Jav);
    if Jav = To'g' then To'g'Jav:= To'g'Jav+ 1;
    writeln;
  end;
{Test natijalarini qayta ishlash}
{to'g'ri javoblar foizini hisoblash}
p:= Round ((To'g'Jav/ JamiSav)*100);
write ('Sizning bahongiz');
case p of
  100:writeln ('A'LO!');
  80..99:writeln ('YAXSHI. ');
  60..79:writeln ('QONIQARLI. ');
  else writeln('YOMON!');
end;
nomer: writeln ('Dastur ishini tugatish uchun <Enter>ni
                bosing');
      readln;
end.

```



{Duymlardan millimetrlarga o'tish jadvalini ekranga, printeriga yoki faylga chiqarish}

var

f:text; {chiqarish fayli}
fname:string; {chiqarish fayli ismi}
dest:integer; {1-ekranga, 2-printeriga, 3-faylga}
d: real; {duymlardagi miqdor}
m: real; {millimetrlardagi miqdor}

begin

```
writeln ('*** Duymlardan millimetrlarga o'tish jadvali ***');
writeln ('Natijani chiqaradi:');
writeln ('1-ekranga;');
writeln ('2-printeriga;');
writeln ('3-faylga.');
```

writeln ('1 dan 3 gacha sonlarni kiriting va <Enter>ni bosing');
write ('Sizning tanlovingiz →');
readln (*dest*);

case *dest* of

- 1: *fname*:=''; {ekranga}
- 2: begin {printeriga}
 fname:= 'pr',;
 write ('Printerni yoqing va <Enter>ni bosing');
 readln;

 end;

- 3: begin {faylga}
 write ('Chiqarish uchun fayl ismini bering →');
 readln (*fname*);

 end;

end;

assign (*f*,*fname*);

rewrite (*f*);

writeln (*f*, '-----');

writeln (*f*, 'Duymlar Millimetrlar');

writeln (*f*, '-----');

d:= 0.5;

while *d* < 10 do

 begin

m:=25.4**d*; {1 duym — 25,4 mm}

 writeln (*f*,*d*:6:1, *m*:10:1);


```

    d:= d+0.5;
  end;
  writeln (f,'-----');
  close (f);
  if dest = 3
  then writeln ('Jadval' ,fname, 'faylida yozilgan');
  writeln ('Dastur ishini tugatish uchun <Enter>ni bosing');
  readln,
end.

```

3.1.13. Rekursiyali dasturlar

192-masala

{«Faktorial» rekursiv funksiyasi}

```

function factorial (k:integer):integer;
begin
  if k = 1
  then factorial:= 1
  else factorial:= k*factorial (k-1);
end;
var
  n: integer; {faktoriali hisoblanishi kerak bo'lgan son}
  f: integer; {n sonining faktoriali}
begin
  writeln ('Faktorialni hisoblash,');
  writeln ('Faktoriali hisoblanishi kerak bo'lgan sonni
    kiriting');
  write ('→');
  readln (n);
  f:= factorial (n);
  writeln (n,'ning faktoriali', f, 'ga teng');
  readln;
end.

```

193-masala

{Ekranga bezaklar chiqaradi}

```

uses Graph, Crt;
{Bezak elementini chizadi}
procedure Elem (x,y, r,p: integer);
  {x, y, r – bezak asosiy elementi markazining
  koordinatalari va radiusi – bezak tartibi}

```

```

begin
  if p >= 0 then
    begin
      Circle (x, y, r); {Radiusi r ga teng, koordinata
                        markazi (x,y) bo'lgan aylana
                        chizadi}
      Delay (100);      {Dastur bajarilishini 100 milli-
                        sekund to'xtatib turadi}
      Elem (x+r, y, Round (r/2), p-1);
      Elem (x, y-r, Round (r/2), p-1);
      Elem (x-r, y, Round (r/2), p-1);
      Elem [x, y+r, Round (r/2), p-1];
    end;
  end;
var
  grDriver:integer;      {drayver}
  grMode:integer;       {grafik rejim}
  grPath:string;        {drayver fayliga yo'l}
  ErrCode:integer;      {grafik rejim xatolik kodi}
begin
  grDriver:=VGA;        {drayver}
  grMode:=VGAHi;       {grafik rejim}
  grPath:= 'e:\tp\bgi'; {drayver joylashgan adres}
  InitGraph (grDriver, grMode, grPath); {Grafik rejimni
  initsializatsiya qiladi. Parametrlar videosistema
  drayverini, ish rejimini, adresni bildiradi}
  ErrCode:= GraphResult; {GraphResult — oxirgi baja-
  rilgan grafik operatsiya natijasini qaytaruvchi
  grafik rejim funksiyasi}
  if ErrCode <> grOK then
    begin
      writeln (' Grafik rejim xatosini initsializatsiyalash. ');
      writeln ('Ishni tugatish uchun <Enter>ni bosing');
      readln;
      Halt (1);
    end;
  {Asosiy dastur}
  Elem (320, 240, 60, 3); {3-tartibli bezak chizamiz}
  OutText ('Dastur ishini tugatish uchun <Enter>ni bosing');
  readln;
end.

```



194-masala

{Rekursiya mexanizmidan foydalanib, n -bo'g'inli elektr zanjir qarshiligini hisoblash}

var

$r1, r2, r3$: real; {zanjir tashkil topgan qarshiliklar kattaligi}

n : integer; {zanjir bo'g'inlari soni (tartibi)}

rz : real; {zanjir qarshiligi}

{ n - tartibli zanjir qarshiligi qiymati}

function Zanj (n : integer): real;

begin

if $n = 1$ then Zanj:= $R1+R2+R3$

else begin

 $rz = \text{Zanj}(n-1)$; $\text{Zanj} := R2 * rz / (R2 + rz)$;

end;

end;

{asosiy protsedura}

begin

writeln ('Elektr zanjir qarshiligini hisoblash.');

writeln ('Qarshilik kattaliklarini kiriting (Om) : ');

write ('r1 →');

readln (r1) ;

write ('r2 →');

readln (r2);

write ('r3 →');

readln (r3) ;

write ('Zanjir tartibi →');

readln (n).;

writeln ('Zanjir qarshiligi:', Zanj(n):6:2, 'Om');

{qarshiliklar kattaligi Zanj protsedurasiga $r1$, $r2$ va $r3$ global o'zgaruvchilar orqali uzatiladi}

readln;

end.



195-masala

{Murakkab elektr zanjir chizmasini chizadi}

{Zanjir chizmasini chizishda rekursiyadan foydalaniladi}

Uses Graph;

const

```

{tur qadami}
dx = 7;
dy = 7;
var
  grDriver:integer;
  grMode:integer;
  grPath:string;
  ErrCode:integer;
  x,y: integer;
  k: integer; {zanjir tartibi}
{k-tartibli zanjir chizmasini chiqaradi}
Procedure Zanj (k: integer; x,y: integer);
begin
  SetColor (Green);
  Line (x, y, x+2*dx, y);
  Rectangle (x+2*dx, y-dy, x+6*dx, y+dy);
  Line (x+6*dx, y, x+8*dx, y);
  OuttextXY (x+3*dx, y-3*dy, 'R1');
  SetColor (Yellow);
  Line (x+8*dx, y, x+8*dx, y+2*dy);
  Rectangle (x+7*dx, y+2*dy, x+9*dx, y+6*dy);
  Line (x+8*dx, y+6*dy, x+8*dx, y+8*dy);
  OuttextXY (x+10*dx, y+2*dy, 'R2');
  SetColor (LightGray);
  Line (x, y+8*dy, x+2*dx, y+8*dy);
  Rectangle (x+2*dx, y+7*dy, x+6*dx, y+9*dy);
  Line (x+6*dx, y+8*dy, x+8*dx, y+8*dy);
  OuttextXY (x+3*dx, y+5*dy, 'R3');
  if k>1 then Cep(k-1,x+8*dx,y);
end;
begin
  grDriver:=VGA;           {drayver}
  grMode:=VGAHi;         {grafik rejim}
  grPath: ='e:\tp\bgi';   {drayver fayliga yo'l}
  InitGraph (grDriver, grMode, grPath); {grafik rejimni
  initsializatsiyalash, parametrlar videosistema drayverini,
  ish rejimini, adresni bildiradi}
  ErrCode:= GraphResult; {oxirgi bajarilgan grafik operatsiya
  natijasini qaytaruvchi grafik rejim funksiyasi}
  if ErrCode <> grOK then

```

```

begin
  writeln ('Grafik rejim xatosining initsializatsiyasi. ');
  writeln ('Ishni tugatish uchun <Enter>ni bosing');
  readln;
  Halt (1)
end;
OutTextXY (10,10, 'Zanjir tartibini kiriting va <Enter>ni
            bosing');
readln (k);
Zanj (k, 10, 50);
readln;
CloseGraph;
end.

```

3.2. YUQORI MURAKKABLIKDAGI MASALALAR ALGORITMLARI VA DASTURLARI

Bu bo'limda dasturdan avval masala yechimining algoritmlari keltiriladi. Bu ba'zan, faqat algoritmik tilda yozilishi qolgan, tugallangan fikr, ba'zi hollarda esa yechim mulohazasi, boshqa hollarda esa dasturni tushuntiruvchi izoh bo'ladi. Lekin hamma hollarda ham dasturni o'qish yengillashtirilgan, o'zlashtirish kerak bo'lgan usullar bayon etiladi. Amallar aniqligi, odatda, doimiy ko'paytuvchi aniqligida ko'rsatilgan.



201-masala

Algoritm. «Ko'ndalang diagonal»

Ikkita son aniqlaymiz: $P = \max(1, 1-k)$ va $g = \min(n, m-k)$. Shunda izlanayotgan miqdor hamma $j = p, p+1, \dots, g$ lar bo'yicha $A[k+j, j]$, elementlar yig'indisiga teng bo'ladi (agar j ning bunday qiymatlari bo'lmasa, ya'ni $p > g$ bo'lsa, u nolga teng). A massivning hamma $m * n$ elementlari qaraladigan yechimni qoniqarli emas, deb qarashi kerak.

Dastur

```

Program Ko'ndalang_diagonal;
const NN=10;
      MM=10;
var
  m,n, j,k,p,g,s :integer;
  A : array [1..MM,1..NN] of integer;

```

```

begin
  writeln ('M,N,K:=');
  readln (m,n,k);
  for p:=1 to m do
    for j:=1 to n do
      begin
        writeln ('A [',P,',',j,']:=');
        readln (A[p,j]);
      end;
    if k>0 then p:=1 else p:=1-k;
    if k+n<m then g:=n
      else
        g:=m-k;
        s:=0;
        for j:=p to g do s:=s+A[k+j,j];
        writeln(s);
      end;
  end.

```



202-masala Algoritm. «Kvadratchalar»

Kvadratchinganin burchaklaridagi hamma sonlar har xil bo'lsa, ularning yig'indisi $0+1+5+1=17$. Lekin buning aksi ham to'g'ri ekanligini ta'kidlash mumkin. Bu dasturni soddalashtiradi.

Dastur

```

Program Kvadratchalar;
const MM=50;
var
  i,j,m,S :integer;
  A : array [1..MM,1..MM] of integer;
begin
  writeln ('M:=');
  readln (m);
  for i:=1 to m do
    for j:=1 to m do
      begin
        writeln ('A [',i,',',j,']:=');
        readln (A[i,j]);
      end;
    S:=0;
  end;

```

```

for i:=1 to m-1 do
  for j:=1 to m-1 do
    if (A[i,j]+ A[i,j+1]+ A[i+1,j]+A[i+1,j+1]=R )
      then S:=S+1;
    writeln (S);
end.

```



203-masala Algoritm. «Sanoq tizimlari»

Berilgan son miqdorini Gornor chizmasi bo'yicha topish kerak:
 $N = ((\dots(M[9]*i + M[8])*i + \dots + M[2])*i + M[1])$
 N , butun sonlar uchun, mumkin bo'lgan maksimal miqdordan oshmaydi, deb faraz qilinadi. N sonlar razryadlari (xonalari) j - tizimda N ni j ga butun bo'lishdagi qoldiqlar sifatida hosil qilinadi:
 $k = N/j$; $M = N - K*j$; $N = K$ va hokazo.
 M ning qoldiqlarini masala shartiga ko'ra, eslab qolish va chop etish shart emas.

Dastur

Program Sanoq_sistemalar;
var

```

i, j, k : integer;
n : real;
m : array [1..9] of integer;

```

begin

```

writeln ('i, j:=');
readln (i, j);
for k:=9 downto 1 do

```

begin

```

write ('M[' , k, ']:=');
readln (m[k]);

```

end;

```

writeln;
n:=0
for k:=9 downto 1 do n :=n*i+m[k];
writeln (trunc(n));
repeat

```

```

write (trunc (n), mod j, "");
n:= trunc(n) div j;

```

```

until n=0;
writeln

```

end.



204-masala
 Algoritm. «Kalendār»

Kabisa bo'lmagan yil oylarida kunlarning son bo'yicha $M[1:11]$ jadvalini hosil qilish qulay. « m » ni « n »ga bo'lishdagi qoldiqni aniqlab beruvchi $MOD(m, n)$ yoki unga o'xshash funksiyani ham ishlatish mumkin.

Dastur

Program Kalendār;

var

a, b, c, i, j : integer;

M : array [1..11] of integer;

function D (x:integer) : boolean;

begin

D:=(c mod x)=0;

end;

begin

writeln ('a,b,c=');

readln (a,b,c);

writeln;

for i:=1 to 11 do

case i of

1,3,5,7,8,10: M[i]:=31;

4,6,9,11: M[i]:=30;

2: M[i]:=28;

end;

j:=a;

for i:=1 to b-1 do j:=j+m[i];

if (b>2) and (D(4) and not D(100) or D(400)) then j:=j+1;

writeln (j);

end.



205-masala
 Algoritm. «So'zning matnga kiritilishi»
 Dastur oson o'qiladi. Izohga hojat yo'q.

Dastur

Program So'zni_matnga_kiritish

const nn=100; kk=50;

var


```

i,n,j,k : integer;
ff : boolean;
x : array [1..nn] of integer;
y : array [1..kk] of integer;
begin
  writeln ('k,n=');
  readln(k,n);
  writeln;
  writeln ('massiv x:');
  for j:=1 to n do readln(x[j]);
  writeln;
  writeln ('massiv y:');
  for j:=1 to k do readln(y[j]);
  writeln;
  i:=0;
  ff:=true;
  while (i<=n-k) and ff do
    begin
      j:=1;
      while j<=k and ff do
        if x[i+j]<>y[j] then ff:=false
        else j:=j+1;
        if ff then
          begin
            writeln ('ha', i+1);
            ff:=false;
          end
        else
          begin
            ff:= true;
            i:=i+1;
          end;
        end;
      if ff then writeln ('yo 'q');
    end.
end.

```



206-masala
 Algoritm. «Nollar seriyasi»
 Dastur oddiy. Izohga hojat yo'q.

Dastur

Program Nollar_seriyasi
 const nn=100;

```

var
    i,n,t,max : integer;
    a : array [1..nn] of real;
begin
    writeln ('n=');
    readln (n);
    for i:=1 to n do readln(a[i]);
    t:=0; max:=0;
    for i:=1 to n do
        begin
            if a[i]<>0 then
                begin
                    if (max<t) then max:=t;
                    t:=0;
                end;
            else t:=t+1;
        end;
    if max< then max:=t;
    writeln;
    writeln (max);
end.

```



207-masala
Algoritm. «Egar nuqta»

Agar m_i i — satrdagi a_{ij} elementlarning eng kichik qiymati, M_j esa j ustundagi a_{ij} elementlarning eng katta qiymati bo'lsa, $m_i \leq a_{ij} \leq M_j$ bo'ladi.

Bu bir nechta mulohazalar qilishga imkon beradi:

a) ixtiyoriy satrning minimumi ixtiyoriy ustun maksimumidan katta emas, ya'ni hamma vaqt $m_i \leq M_j$ shart o'rinli;

b) agar qandaydir i va j uchun $m_i = M_j$ tenglik bajarilsa, minimumlarning eng kattasi maksimumlarning eng kichigi bilan mos keladi, i satrning j ustun bilan kesishishida esa egar nuqta:

$m_i = a_{ij} = M_j$ yotadi; (*)

d) agar a_{ij} — egar nuqta bo'lsa, uning uchun (*) shart bajariladi va, demak, minimumlarning maksimumi, maksimumlarning minimumiga teng bo'ladi.

Aytilganlarni dasturga joriy etish kerak. Har bir satrda minimum m_i izlanadi, bu minimumlar ichidan maksimum — M_a tanlanadi

va u joylashgan i_0 satr eslab qolinadi. Agar minimum «Ma» ning joriy qiymatidan kichik bo'lib qolsa, satrda minimumni izlashni to'xtatish kerak.

Keyin, maksimal element topilgan «Ma» qiymatga teng ustun izlanadi, agar bunday ustun bo'lmasa, egar nuqta ham yo'q. Amalda dasturda, agar ustunda Ma elementdan katta element bo'lmasa, a) punktga ko'ra, ustun, izlanayotgan ustun ekanligi ko'rsatiladi.

Agar dasturda bu yerda keltirilgan mulohazalardan foydalanilmasa, u ancha sekin ishlaydigan bo'ladi.

Dastur

```
Program Egar_nuqta.  
const nn=20; mm=20;  
label 1,2;  
var  
    i,n,j,m,io,mi,ma :integer;  
    a : array [1..mm,1..nn] of integer;  
begin  
    writeln ('m,n=');  
    readln (m,n);  
    for i:=1 to m do  
        begin  
            writeln ('a[i,j]=');  
            readln (a[i,j]);  
        end;  
    for i:=1 to m do  
        begin  
            if (i>1) and (a[i,j] <=ma) then goto 1;  
            if (j=1) or (a[i,j] <mi) then mi:=a[i,j];  
1 : end;  
            ma:=mi; io:=i;  
            for i:=1 to n do  
                begin  
                    for i=1 to m do if a[i,j]> ma goto 2;  
                    writeln (io:3,j:3); exit;  
                end;  
            2: end;  
            writeln(0);  
        end;  
end.
```



208-masala
 Algoritm. «Kasrni qisqartirish»

Yevklid algoritmini dasturlash mumkin. Unga ko'ra m va n sonlarning eng katta umumiy bo'luvchisini topish va unga qisqartirish kerak.

Yevklid algoritmi mohiyatini tushuntiramiz, $m1 \leq m2$ bo'lsin, $(m1, m2)$ juftlikning ixtiyoriy umumiy bo'luvchisi $(m2, m1 - m2)$ juftlikning, demak, $(m2, m3)$ juftlikning ham umumiy bo'luvchisi bo'ladi, bu yerda,

$$m3 = m1 - (m1 / m2) * m2$$

$m1$ ni $m2$ ga bo'lishda qoldiq bo'ladi, shuning uchun $m3 < m2$ oldindan ma'lum. Aksi ham to'g'ri: $(m2, m3)$ juftlikning har qanday umumiy bo'luvchisi (u.b.) $(m1, m2)$ juftlikning ham umumiy bo'luvchisi bo'ladi. Shuning uchun

$$u.b. (m1, m2) = u.b. (m2, m3)$$

$u.b. (m1, m2)$ funksiyada argumentni uni kichigiga bo'lgandagi qoldig'i bilan ketma-ket almashtira borib,

$$u.b. (m1, m2) = u.b. (m2, m3) = \dots = u.b. (mk, 0) = mk$$

ketma-ketlik hosil qilinadi, unda $m1 \geq m2 > \dots > mk > 0$ va mk boshlang'ich $m1, m2$ sonlarning eng katta umumiy bo'luvchisi bo'ladi.

Agar n katta son bo'lmasa, j -eng kichik qiymatli $i/j = m/n$ kasrni tanlab olish mumkin. U, albatta, qisqarmaydi. Faqat kasrlar tengligini nisbatlar tengligi bilan emas, ko'paytmalar tengligi bilan tekshirish kerak:

$$i * n = j * m$$

Bu taklif etilgan dasturga olib keladi.

Dastur

```

Program Kasrni_qisqartirish;
label 1;
var
  i,n,j,m :integer;
begin
  readln (m,n);
  writeln ('m=',m,'n=',n);
  for j:=1 to n do
    begin
      i:=j*m div n;
      if i*n=j*m then goto 1;
    end
  end
  
```

```

        end;
1: writeln ('m/n',i,'/',j);
end.

```



209-masala

Algoritm. «Massivlarning qo'shilishi»

Bu muhim masala $m+n$ amalda bajarilishi kerak. A va B dan birinchi elementlarini olamiz, ulardan kichigini C massivga yozamiz va uni o'z massividan navbatdagisi bilan almashtiramiz. Yana ikkitadan kichigini tanlab, keyin C ga kiritamiz va hokazo, har bir taqqoslashdan keyin C ga element qo'shiladi, demak, taqqoslash $m+n$ dan kichik bo'ladi. Faqat dasturning, massivlardan bittasi tamom bo'lganidan keyin ham, to'g'ri ishlashini ta'minlash kerak.

Dastur

```

Program Massivlar_qo'shilishi;
const MM=100;
      NN=100;
      MN=200;
var
  m,n,i,j,k :integer;
  A : array [1..MM] of integer;
  B : array [1..NN] of integer;
  C : array [1..MN] of integer;
begin
  writeln ('M,N:=');
  readln (m,n);
  writeln ('array A:=');
  for i:=1 to m do readln(A[i]);
  writeln ('array B:=');
  for j:=1 to n do readln(B[j]);
  i:=1; j:=1;
  for k:=1 to m+n do
    begin
      if ((i>m) or (A[i]>B[j])) and not
        (j>n) then
        begin
          c[k]:=B[j]; j:=j+1;
        end;
    end;
  else

```

```

begin
    c[k]:=A[i]; i:=i+1;
end;
end;
writeln ('array C:=');
for k:=1 to m+n do writeln (c[k]);
end.

```



210-masala Algoritm. «Moda»

Boshlang'ich massivni tartibga keltirish, shundan keyin bir qarashda qiymatlar chastotasini — eng ko'p uchraydigan qiymatni hisoblash kerak. Chastotani yozish uchun massiv tashkil etishga hojat yo'q, tez uchraydiganlar nomzodini eslab qolish va uning chastotasini qaralgan elementning topilgan chastotasi bilan taqqoslash yetarli.

Agar qandaydir holga ko'ra, tartiblashtirishni bajarmaslik kerak bo'lsa, quyidagicha yo'l tutiladi: navbatdagi $a[i]$ (boshida bu $a[1]$ element) undan keyin keladigan element bilan taqqoslanadi. Bunda, birinchidan, $a[i]$ qiymatning paydo bo'lish chastotasi hisoblanadi, ikkinchidan, $a[i]$ ga teng bo'lgan $a[j]$ elementlari ketma-ket $a[i+1]$, $a[i+2]$... elementlari bilan almashtiriladi. A massiv oxirigacha qarab chiqilib, $a[i]$ element chastotasi aniqlanganidan keyin $a[i+k]$ element chastotasi topiladi, $a[i]$ va $a[i+k]$ o'rtasidagi elementlar o'tkazib yuboriladi. Bu algoritm samarali ko'rinadi va agar massivda har xil elementlar bo'lsa, $n*n$ ta amal bajarilishi kerak bo'ladi. Bu fikrlar taklif etilgan dasturda amalga oshirilgan.

Dastur

```

Program Moda;
const NN=100;
var
    n,m,am,i,j,k : integer;
    A : array [1..NN] of integer;
begin
    writeln ('n:=');
    readln (n);
    for i:=1 to n do readln(a[i]);
    writeln;

```

```

m:=0; i:=1;
while i+m<=n do

```

```

    begin
        k:=1;
        for j:=i+1 to n do
            if a[j]=a[i] then
                begin
                    a[j]:=a[i+k]; k:=k+1;
                end;
            if m<k then
                begin
                    am:=a[i]; m:=k;
                end;
            i:=i+k;
        end;

```

```

    writeln (am);
end.

```



211-masala

Algoritm. «Markaziy qishloq»

Bu masalani quyidagicha bayon etamiz: A massivning har bir i satrida $a[i,j]$ ($j \neq i$) sonlari ichidan eng kattasi tanlanadi va u $a[i,j]$ ga qo'shiladi. Tegishli yig'indini eng kichik qiluvchi i topilishi kerak. Masalaning bunday qo'yilishida uning standart yechimga egaligiga yo'l qo'yiladi.

Dastur

```

Program Markaziy_qishloq;

```

```

const KK=20;

```

```

var

```

```

    i,j,k,i1,s,t :integer;

```

```

    A : array [1..KK,1..KK] of real;

```

```

begin

```

```

    writeln ('k:=');

```

```

    readln (k);

```

```

    for i:=1 to k do

```

```

        for j:=1 to k do

```

```

            begin

```

```

                writeln('a['i','j,']:=');

```

```

                readln(a[i,j]);

```

```

            end;

```

```

    for i:=1 to k do
      begin
        s:=0;
        for j:=1 to k do
          if (j<>) and (s<a[i,j]) then s:=a[i,j];
        s:=s+a[i,j];
        if (i=1) or (s<t) then
          begin
            i1:=i;
            t:=s;
          end;
        end;
      writeln (i1);
    end.

```



212-masala
 Algoritm. «Tartib indekslari»

Hammasidan osoni «izlanayotgan tartib indekslari» uchun qo‘shimcha $I[1:N]$ massiv hosil qilish, avval uni $1, 2, \dots, N$ sonlari bilan to‘ldirish, keyin berilgan A massivni, qandaydir usul bilan tartibga keltirish, xuddi shunday tartibda I massiv elementlarining o‘rmini almashtirish kerak. A tartibga keltirilgan vaqtda, I izlanayotgan massiv bo‘ladi.

Keltirilgan dasturda xuddi shu ish bajarilgan, bu yerda tartibga keltirishning eng oddiy usuli ishlatilgan: A massivda eng kichik element bor va bu element birinchisi bilan o‘rmini almashtirib boradi. Keyin bu jarayon $A[2:N]$ kesmaga va hokazo qo‘llaniladi.

Agar dasturni murakkablashtirish kerak bo‘lsa, I massivsiz ishini bajarish mumkin, lekin A massiv elementlari o‘zgarmasligi kerak. Bunda navbatdagi minimal element (MI) indeksini chop etish va eslab qolish yetarli. Buning uchun, albatta, birinchi shunday elementning indeksini va birinchi maksimal element indeksini oldindan topib qo‘yish kerak.

Dastur

```

Program Tartib_indekslari;
const NN=100;
label 1,2;
var

```



```

j,i,m,mi,ma,k,n :integer;
A : array [1..NN] of real;
begin
  Writeln('N:=');
  readln(n);
  for i:=1 to n do
    begin
      writeln ('a[' ,i ,']:=');
      readln (a[i]);
    end;

  mi:=1; ma:=1;
  for k:=1 to n do
    begin
      if a[k]<a[mi] then mi:=k;
      if a[k]<a[ma] then ma:=k;
    end;
    writeln (mi);
    for m:=2 to n do
      begin
        i:=ma;
        for k:=1 to n do
          begin
            if (a[k]<a[mi]) or (a[k]=
a[mi]) and (k<=mi) then goto 1;
            if a[k]=a[mi] then
              begin
                i:=k;
                goto 2;
              end;
            if a[k]<a[i] then
              i:=k:
            1 : end;
            2 : mi:=i;
            writeln (mi);
          end;
        end;
      end;
    end.

```



213-masala
 Algoritm. «Nollashtirish»

Masalani yechish uchun eng oson yo‘l $B[1:m]$ va $C[1:n]$ ikkita yordamchi massivlarni kiritish. Keyinchalik A massiv elementlari

ko'zdan kechirila borib, B va C massivlarda nollar uchragan satr va ustunlar belgilab boriladi, ya'ni: $\text{if } a[i,j] = 0 \text{ then } b[i] = 1 : c[j] = 1$ bajariladi.

Endi, A massiv ikkinchi marta ko'zdan kechirilayotganda, B yoki C massivlarda belgilarga ega bo'lgan, satr va ustunlarda joylashgan elementlar nollar bilan almashtiriladi, ya'ni:

$\text{if } b[i] = 1 \text{ or } c[j] = 1 \text{ then } a[i,j] = 0.$

Faqat bitta $C[1:n]$ yordamchi massivdan foydalanadigan boshqa yechimni taklif qilish mumkin. Bu holda A massiv satrlar bo'yicha ko'zdan kechiriladi. Nol elementlari bo'lgan ustunlar, avvalgiday C massivda belgilanadi, ko'rilayotgan satrda nol uchragani esa Z o'zgaruvchida qaraladi, ya'ni:

$\text{if } a[i,j] = 0 \text{ then } c[j] = 1 : Z=1.$

i catr qarab chiqilgach, agar $z=1$ bo'lsa, satr nollashtiriladi. Ish so'ngida C massiv qaraladi va agar $c[j] = 1$ bo'lsa, j ustunlar nollashtiriladi.

Bu masalani yechishga xos xatoliklar:

1) A massiv qaraladi va agar $a[i,j]$ element nolga teng bo'lsa, i satr yoki j ustun yoki unisi ham, bunisi ham nollashtiriladi. Keyingi qarashlarda nollashtirilgan elementlar boshlang'ich nolga teng element, deb qabul qilinadi va ortiqcha nollashtirishlarni hosil qiladi.

2) A massiv qaraladi va agar $a[i,j]=0$ bo'lsa, navbatdagi $c[k]$ elementga j kiritiladi. Bundan bitta j raqam ko'p marta kiritilishi va C massiv, shartga zid ravishda, $m*n$ o'lchamda talab qilinishi mumkin.

Dastur

Program Nollashtirish;

const NN=20;

MM:=20;

var

i,j,m,n :integer;

Z : boolean;

A : array [1..MM,1..NN] of real;

C : array [1..NN] of boolean;

begin

writeln ('M,N:=');

readln (m,n);

for i:=1 to m do

for j:=1 to m do

```

begin
  write ('a[',i,',',j,']:=');
  readln (a[i,j]);
end;
for j:=1 to n do C[j]:=false;
for i:=1 to m do
begin
  z:=false;
  for j:=1 to n do
    if a[i,j]=0 then
      begin
        z:=true;
        c[j]:=true;
      end;
  if z then for j:=1 to m do a[i,j]:=0;
end;
for j:=1 to n do
  if c[j] then for i:=1 to m do a[i,j]:=0;
for i:=1 to m do
begin
  for j:=1 to n do writeln(a[i,j]:4);
  writeln;
end;
end.

```



214-masala
 Algoritm. «Ulgurji xarid»

Ulgurji xarid orttirmalarsiz quyidagicha topiladi:

$$n1 = n \quad 144: m = n - n1 * 144$$

$$n2 = m \quad 12: n3 = m - n2 * 12$$

uni faqat ikki yo'l bilan arzonlashtirish mumkin. Ortiqcha bog'lam olib, ortiqcha juftlik olmaslik yoki ortiqcha quti olib, bog'lam ham, juftlik ham olmaslik kerak. Agar $n3 * 1.05 > 10.25$ bo'lsa ortiqcha:

$$n2 = n2 + 1; \quad n3 = 0$$

bog'lam olish kerak.

Agar hosil bo'lgan xaridda (eski yoki o'zgartirilganida) $n2 * 10.25 + n3 * 1.05 > 114.00$ bo'lsa, $n1 = n1 + 1; \quad n2 = 0; \quad n3 = 0$ ni hosil qilish kerak.

Bu yechim har xil narxlarda (quti 12 bog'lamdan, bog'lam 12 juftdan arzon bo'lgan hollarda) yaroqlidir.

Test misollar

n	n1	n2	n3
9	0	0	9
10	0	1	0
131	0	11	0
134	1	0	0

Dastur

```
Program Ulgurji_savdo;
var
    n1,n2,n3,m,n :integer;
begin
    writeln ('N:=');
    readln (n);
    n1:=n div 144;
    m:=n-n1*144;
    n2:=m div 12;
    n3:=m-n2*12;
    if (n3*1.05 )>10.25 then
        begin
            n2:=n2+1;
            n3:=0
        end;
    if (n2*10.25 + n3*1.05 )>114 then
        begin
            n1:=n1+1;
            n2:=0;
            n3:=0
        end;
    writeln (n1,' ',n2,' ',n3 );
end.
```



215-masala

Algoritm. «To'ntarilgan sonlar»

Bu masala (N^2 amalda bajarilish uchun) ma'lum bir fikrlashni talab etadi. Tadqiq qilinadigan kesmani uning markazi bilan berish (bu N ta variantni beradi) va markazdan ikkala tomonga simmetrik elementlarni taqqoslash kerak (bu N ta guruhdan katta bo'lmagan amallarni talab qiladi). Shunday qilib, yechim N^2 amalda (doimiy

ko'paytuvchi aniqligigacha) topiladi. Yechimni rasmiylashtirishda kesma markazi o'rniga (u elementga ham, elementlar o'rtasiga ham to'g'ri kelishi mumkin) markaz bilan qo'shni bo'lgan ikki element raqamini $LN < PN$ olish qulay. Bu raqamlar tushunarli tarzda siljiydi:

$$(LN, PN) = (1, 2), (1, 3), (2, 3), (2, 4), \dots$$

taqqoslanadigan element raqamlari boshda $L=LN : P=PN$, deb olinadi va keyin taqqoslanadigan $A[L]$ va $A[P]$ elementlar teng bo'lgunicha yoki L va P nuqtalardan bittasi $(1, N)$ chegaradan chiqquncha, har qadamda $L=L-1 : P=P+1$ bajariladi. Shunda (LN, LP) markazi maksimal kesma $M=P-L-1$ uzunlikda bo'ladi.

Dastur

```

Program To'ntarilgan_sonlar;
const NN=100;
var
  l,i,ln,pn,p,m,max,n:integer;
  z : boolean;
  A : array [1..NN] of real;
begin
  writeln ('n:=');
  readln(n);
  for i:=1 to n do
    begin
      writeln ('a['i',']:=');
      readln (a[i]);
    end;
  max:=1; z:=true; ln:=1; pn:=2;
  while 2*(n*pn+1)+1>max do
    begin
      l:=ln; p:=pn;
      while (l>=1) and (p<=n) and (A[l]=A[p]) do
        begin
          l:=l-1; p:=p+1;
        end;
      m:=p-l-1;
      if (max<m) then max:=m;
      if z then pn:=pn+1 else ln:=ln+1;
      z:=NOT z;
    end;
  writeln (max);
end.

```



$i=1, j=n$ deb olinadi, $a[i,j]$ element (avval bu, keyin navbatdagisi) x soni bilan taqqoslanadi.

agar $a[i,j] = x$ bo'lsa, javob topiladi,

agar $a[i,j] < x$ bo'lsa, $i=i+1$ deb olinadi,

agar $a[i,j] > x$ bo'lsa, $j=j-1$ deb olinsin.

Oxirgi ikki holda $i \leq m$ va $j \geq 1$ qolganligini tekshirish kerak. Agar qolgan bo'lsa, taqqoslashga qaytish, aks holda «yo'q» deb chop etish kerak. Har bir qadamda i oshirilib, yoki j kamaytirilib boriladi. Demak, qadamlar $m+n$ dan ko'p bo'lmaydi.

Bu algoritmning to'g'riligiga ishonch hosil qilish qiyin emas. Haqiqatan ham, agar $a[i,j] < x$ bo'lsa, x soni $a[i:m, 1:j]$ massivning i satrida yo'q va bu satrni tashlash mumkin. Agar $a[i,j] > x$ bo'lsa, xuddi shu sababga ko'ra j ustunni tashlash mumkin.

Dastur

```
Program Ikki_marta_monoton;
```

```
const MM=20;
```

```
      NN=20;
```

```
label l;
```

```
var
```

```
    i, j, m, n, x :integer;
```

```
    A : array [1..MM,1..NN] of real;
```

```
begin
```

```
    writeln ('m,n:=');
```

```
    readln (m,n);
```

```
    writeln ('x:=');
```

```
    readln (x);
```

```
    for i:=1 to m do
```

```
        for j:=1 to n do
```

```
            begin
```

```
                writeln ('a['i','j',']:=');
```

```
                readln (a[i,j]);
```

```
            end;
```

```
    i:=1; j:=n;
```

```
    while (i<=m) and (j>=1) do
```

```
        begin
```

```
            if a[i,j]=x then goto l;
```

```
            if a[i,j]<x then i:=i+1 else j:=j-1;
```

```

        end;
        writeln ('Yo'q');
1: writeln (i, ' j');
end.

```



217-masala Algoritm «Tetraedrlar»

Yechim ikki qismga bo'linadi: $M1$ bilan mos keltirish uchun N tetraedrning N' qirrasini va $M1$ da qoldiradigan N tetraedrning burilishlarini tanlash. Birinchisi to'rt usul bilan, ikkinchisi uch usul bilan bajariladi. Faqat, N tanlanganidan keyin N ning qolgan qirralarini birontasidan boshlab ma'lum tartibda (masalan, soat yo'nalishiga teskari) yozib olish, yechimning ikkinchi qismini funksiya ko'rinishida rasmiylashtirish kerak (dasturda keltirilgan).

Yechimdan hamma 12 variant bevosita yozib olinadigan javob qoniqarli emas.

Dastur

Program Tetraedrlar;

var

$m1, m2, m3, m4, n1, n2, n3, n4$:integer;

function DA (i,j,k,l:integer) : boolean;

begin

DA:=((m1=i) and (((m2=j) and (m3=k) and (m4=i))

or ((m2=1) and (m3=j) and (m4=k))

or ((m2=k) and (m3=1) and (m4=j))))

writeln ('m1,m2,m3,m4:=');

readln (m1,m2,m3,m4);

writeln ('n1,n2,n3,n4:=');

readln (n1,n2,n3,n4);

writeln;

if DA (n1,n2,n3,n4) or DA(n2,n1,n4,n3) or DA(n3,n1,n2,n4) or

DA(n4,n1,n3,n2) then writeln('Ha')

else writeln('Yo'q');

end.



218-masala Algoritm. «Kasr davri»

Yechim shartda ko'rsatilmagan qo'shimcha talablarga qattiq bog'liq. « M » natural sonni « N » natural songa bo'lganda bo'linmaning butun qismi va qoldiq hosil bo'ladi:

$$i=M/N : k=M-i*N.$$

Masalani yechish uchun avval M/N kasrdan M suratni qoldiqqa almashtirib, butun qismini yo'qotamiz:

$$M= M-M/N*N.$$

Endi, hosil bo'lgan kasrning i bo'linma va M qoldiq raqamlarini ketma-ket hosil qilish mumkin:

$$i= 10*M/N : M=10*M-i*N \text{ va hokazo.}$$

Har bir qoldiq N dan oshmaydi. Demak, har xil qoldiqlar N dan katta bo'lmaydi va ular takrorlana boshlaydi. Qoldiq qachon takrorlansa, bo'linmalar ham takrorlana boshlaydi, davr boshlanadi. Takrorlashni ilg'ab olish uchun, eng osoni $D[1:N]$ massivni tashkil qilish va unga, hosil bo'lish tartibiga ko'ra, qoldiqlarni yozish va har bir yangi qoldiqni barcha oldingilari bilan taqqoslab borish kerak.

Bayon etilgan yechim to'g'ri, lekin yaxshi emas. Hisob bo'yicha K — qoldiqqa K taqqoslash, hamma qoldiqlarga esa $N^2/2$ taqqoslash ketadi. Zamonaviy EHM larda 10^8 tartibli butun M va N ni hosil qilish oson, lekin 10^{16} amallarni bajarish uchun yillar ketadi, shu sabab N soni uchun D massivni tashkil qilishga kelishilgan ekan, uni tozalaymiz va navbatdagi M qoldiqning paydo bo'lishini $D[M]$ elementda belgilaymiz. U paytda K qoldiq «yangiligi»ni tekshirish bitta taqqoslashni egallaydi.

Lekin bu yechim ham jiddiy kamchilikka ega. N soni shunday katta bo'lishi mumkinki, bunda N ta amalni bajarish mumkin bo'ladi-yu, lekin N ta elementdagi massivni tezkor xotiraga kiritish mumkin bo'lmay qoladi. Qoldiqlar uchun massivni tashkil qilmay, davrni topishga urinib ko'ramiz. Dastlab avvalgi kabi « M » dan « N » ga bo'linadigan qismini ajratamiz. Keyin bo'linmaning N raqamini o'tkazamiz. Endi, davr boshlangani ma'lum bo'lgach, bitta yagona qoldiqni eslab qolamiz va bo'linmaning raqamlari takrorlana boshlanmagunicha chop etamiz. Bu ishlar taklif etilgan dasturda keltirilgan. Bu dastur oldingisidan qisqa.

Dastur

```
Program Kasr_davri;  
var  
    m,n,i,j,k :integer;  
begin  
    writeln ('m,n:=');  
    readln (m,n);
```



```

writeln;
M:=m-(m div n)*n;
k:=1;
while (k<=n) or (j<>m) do
begin
if k=n then j:=m;
i:= 10 * m div n;
m:=10 * m-i * n;
if k>=n then write(i);
k:=k+1;
end;
writeln;
end.

```



219-masala
 Algoritm. «Arra»

Masalani tushuntirish talab qilinmaydi.

Dastur

```

Program Arra;
const mm=100;
var
  i,k,j,m :integer;
  x : array [1..mm] of real;
begin
  writeln ('m:=');
  readln (m);
  for i:=1 to m do readln(x[i]);
  j:=1; k:=1; i:=1;
  while i+1<m do
begin
  if (x[i]<x[i+1]) and (x[i+1]>x[i+2]) then
begin
  i:=i+2; j:=j+2;
  if k<j then k:=j
end
else
begin
  i:=i+1; j:=1;
  if i+k>=m then i:=m-1;
end;
end;
writeln;
writeln (k);
end.

```



220-masala
 Algoritm. «Nollar oxiriga»

Masala ikki siklni kiritish bilan oson yechiladi: birinchisi nolga teng bo'lmagan elementlarni massiv boshiga yozadi, ikkinchisi esa qoldiqni nollar bilan to'ldiradi. Lekin bu sikllarni birgalikda ham tashkil etish mumkin.

Dastur

```

Program Nollar_oxiriga;
const nn=100;
var
  i,n,j      : integer;
  x : array [1..nn] of integer;
begin
  write ('n=', n);
  readln (n);
  for i:=1 to n do readln(x[i]);
  j:=0;
  for i:=1 to n do
    if x[i] <>0 then
      begin
        j:=j+1;
        if i<>j then
          begin
            x[j]:=x[i]; x[i]:=0;
          end;
        end;
      end;
  for i:=1 to n do writeln(x[i],");
  writeln;
end.
  
```



221-masala
 Algoritm. «Har xil raqamli sonlar»

Dastur kerakli darajada tushunarli.

Dastur

```

Program Har_xil_raqamli_sonlar;
var
  i,m,j,k:integer;
begin
  
```

```

for i:=1 to 9 do
  for j:=0 to 9 do if i<>j then
    for k:=0 to 9 do if (k<>i) and (k<>j) then
      for m:=0 to 9 do
        if (m<>i) and (m<>j) and (m<>k) then
          writeln (((i10+j)*10+k)*10+m);
end.

```



222-masala

Algoritm. «0,1,2 larning o'rmini almashtirish»

Bu masalani yechishda hech nimaning o'rmini almashtirish kerak emas. Massivda nechta nollar, birlar va ikkilar borligini sanash va massivni talab bo'yicha to'ldirish kerak.

Dastur

Program O'rin_almashtirish;

const nn=100;

type el=0..2;

var

i,n : integer;

x : array [1..nn] of el;

a : array [e1] of 0..nn;

begin

readln (n);

writeln ('n:=',n);

for i:=1 to n do readln (x[i]);

writeln;

a [0]:=0; a[1]:=0;a[2]:=0;

for i:=1 to n do a[x[i]]:=a[x[i]]+1;

for i:=1 to n do

if i<=a[0] then x[i]:=0

else if i<=n-a[2] then x[i]:=1 else x[i]:=2;

for i:=1 to n do writeln (x[i]);

end.



223-masala

Algoritm. «Arifmetik amallar»

Bu masala ham barcha o'rin almashtirishlarni, faqat takrorlash bilan hosil qilishga bag'ishlangan. Berilgan ifodani quyidagi ko'rinishda yozamiz:

$w = (((1 a_2 2) a_3 3) a_4 4) a_5 5) a_6 6$

bu yerda, $A[6]$ massiv elementlari arifmetik amal belgilarini bildiradi. $+, -, *, \div$ belgilar 1,2,3,4 qiymatlar bilan ifodalanadi, deb kelishib olamiz. Variantlar ko'p bo'lishi mumkinligi uchun ($4^5 > 1000$), hisoblashlarni tejab tashkil qilishga harakat qilamiz. Buning uchun, w ni hisoblab borib, oraliq natijalarni $B[6]$ massiv elementlari ($B_1 := 1; B_2 := B_1 a_2 2: \dots : B_6 := B_5 a_6 6$ va $w := B_6$) deb olib xotirlaymiz. Keyin variantlarni ko'rib chiqq borib, avval $a_6 = 1, 2, 3, 4$ ni almashtiramiz. Har bir bunday o'zgarishda faqat B_6 ni hisoblab chiqish kerak bo'ladi. Keyin $a_5 = a_5 + 1$ ni oshiramiz, B_5 ni qayta hisoblab, $a_6 = 1, 2, 3, 4$ va hokazo.

Dastur

```

Program Arifmetik_amallar;
const m=1; n=9;
label 5;
var
    j, k, y :integer;
    A,B : array [1..n] of integer;
begin
    b[1]=1; k:=0; a[2]:=0;
    for i:=3 to n do a[i]:=4
5: for i:=n downto 2 do
    if (a[i]=4) then a[i]:=1
        else
            begin
                A[i]:=A[i]+1;
                Y:=B[i-1];
                case (A[i]) of
                    1:B[i]:=y+i;
                    2:B[i]:=-y-i;
                    3:B[i]:=y*i;
                    4:B[i]:=y div i;
                end;
                for j:=j+1 to n do b[j]:=b[j-1]+j;
                if (b[n]<>M) then goto 5; k:=k+1;
                for j:=2 to n-1 do write ((')); write ('1');
                for j:=2 to n do
                    begin
                        if (j>2) then write (('))
                        case a[j] of

```

```

1: write ('+');
2: write ('-');
3: write ('*');
4: write ('%');
   end;
   writeln (j);
   end;
writeln ('=',m); goto 5;
end;
writeln(k);
end.

```



224-masala
 Algoritm. «Tez daraja»

Odatdagi a^k ni hisoblashda dastlab qiymati 1 ga teng b o'zgaruvchi kiritiladi va ko'p marta

$$K:=K-1; b:=b*a$$

operatorlar bajariladi.

K o'zgaruvchi nolga teng bo'lgan paytda (bu k siklni talab qiladi), b izlanayotgan a^k miqdorga teng bo'ladi. Hisoblashni qisqartirishning mohiyati quyidagicha. Dastlab, $b:=1$. Agar K toq bo'lsa, avvalgiday

$$K:=K-1; b:=b*a$$

operatorlar bajariladi.

Agar k juft bo'lsa, unda,

$$a^k = (a^2)^{k/2}$$

ayniyatdan foydalanib, $k:=k/2$; $a:=a*a$ almashtirish bajariladi.

Nihoyat, « K » o'zgaruvchi nolga teng bo'lgan paytda, « b » o'zgaruvchi izlangan miqdorga teng bo'ladi. Isbot uchun izlanayotgan a^k miqdorni w deb belgilaymiz, unda

$$b:=1; a^k \cdot b=w \quad (*)$$

agar K toq bo'lsa,

$$K:=K-1; b:=b \cdot a$$

shakl almashtirishni bajarib, (*) tenglikni buzmaymiz. Agar k juft bo'lsa:

$$K:=K/2; a:=a \cdot a$$

almashtirish ham uni buzmaydi. K nolga teng bo'lgan paytda, (*) tenglik

$$a^0 \cdot b=w, \text{ ya'ni } b=w,$$

tenglikka aylanadi va, demak, b izlanayotgan miqdorga teng bo'ladi.

Dastur

```
Program Tez_daraja;  
var  
  a, b : real;  
  k, n : integer;  
begin  
  readln (a,k);  
  write(a, o'^o', k, o'=o');  
  b:=1;  
  while k>0 do  
    begin  
      n:=k div 2; if (n+n<k) then b:=b*a;  
      k:=n; a:=a*a;  
    end;  
  writeln (b);  
end.
```



225-masala

Algoritm. «Har xil sonlar»

Dastur oson o'qiladi.

Tekshirilayotgan element birinchi takrorlashgacha oldingisi bilan emas, balki keyingisi bilan taqqoslansa, dastur o'qilishi yana ham tezlashadi. Bundan takrorlanishlar soni $m*m$ dan $m*k$ gacha tushadi. Bu yerda, massivda m — hamma sonlar, k — esa har xil sonlar miqdorini bildiradi.

Dastur

```
Program Har_xil_sonlar;  
const n=100;  
var  
  i, j, m, S : integer;  
  b : boolean;  
  A : array [1..n] of integer;  
begin  
  readln (m);  
  writeln ('m:= ',m);  
  for i:=1 to n do  
    begin  
      readln (a[i]);  
      writeln ('a[',i, ']:= ',a[i]);  
    end;
```

```

S:=0;
for i:=1 to m do
  begin
    b:=false;
    j:=i+1;
    while (j<=m) and not b do
      begin
        b:=b or (a[i]=a[j]);
        j:=j+1;
      end;
    if not b then S:=S+1;
  end;
writeln ('S=',S);
end.

```



226-masala

Algoritm. «Minimumlarning maksimumi»

Keltirilgan dastur, birinchi satr va satrning har bir elementi alohida ko'zdan kechirishni talab qilinmaydigan qilib tuzilgan. Bundan tashqari, satrni ishlab chiqish navbatdagi satr izlanayotgan elementga ega emasligi ma'lum bo'lishi bilan to'xtatiladi.

Dastur

Program Minimumlar_maksimumi;

const mm=10; nn=20;

label 1;

var

i, j, n, m, k :integer;

min, max:integer;

x : array [1..mm,1..nn] of integer;

begin

readln (m,n);

writeln ('m:= ',m, 'n:= ',n);

for i:=1 to m do

for j:=1 to n do

begin

writeln (x[i,j]);

readln (x[i,j]);

end;

```

for i:=1 to m do
  begin
    for j:=1 to n do
      begin
        if (i>1) and (x[i,j])<=max) then goto 1;
        if (j=1) or (x[i,j])<=min) then min:= x[i,j];
      end;
    max:=min; k:=i;
  1 : end;
  writeln ('k',k);
end.

```



227-masala Algoritm. «Spiral»

Spiral tarmog'i to'rtta to'g'ri chiziqli qismga bo'linadi: chapdan o'ngga gorizonta, yuqoridan pastga vertikal, o'ngdan chapga gorizonta va quyidan yuqoriga vertikal. Dasturda ularning har biri alohida to'ldiriladi va bir qismning ikkinchisiga «sirpanib o'tishi» ishlatiladi. Uning tugashi oxirgi $k=n*n$ sonining kiritilishi bilan tekshiriladi.

Dastur

```

Program Spiral;
const nn=19;
var
  i, n, j, k : integer;
  function MOV :boolean;
  A : array [1..nn,1..nn] of integer;
begin
  MOV:=false;
  if k<=n*n then
    begin
      a[i,j]:=k; k:=k+1;
      MOV:=true
    end;
end;
begin
  writeln ('n:=',n);
  readln (n);
  k:=1; i:=1; j:=1;

```



```

repeat
    while MOV and (i+j<n+1) do j:=j+1; k:=k-1;
    while MOV and (i<j) do i:=i+1; k:=k-1;
    while MOV and (i+j<n+1) do j:=j-1; k:=k-1;
    while MOV and (i>j+1) do i:=i-1; k:=k-1;
until k:=n*n;
for i:=1 to n do
    begin
        for j:=1 to n do writeln ( a[i,j]:4);
        writeln;
    end;
end.

```



228-masala

Algoritm. «Ichki to'plam bo'yicha yig'indi»

Bu masala hamma ichki to'plamlarni sanab o'tishga doir. Tashqaridan qaraganda u variantlarni ko'rib chiqish masalasiga o'xshaydi, lekin unga qaraganda ancha sodda, b — natural son va b_i — uning ikkilik xonalari ($b_i=0$ yoki $b_i=1$) bo'lsin:

$$b = b_1 + 2b_2 + \dots + 2^{n-1} * b_n.$$

$b = 1, 2, \dots, 2^{n-1}$ qiymatlar ketma-ketligi bo'ylab « b » o'tganda, birga teng b_i elementlarning i indekslar to'plami $\{1, 2, \dots, n\}$ to'plamning hamma (bo'sh bo'lmagan) ichki to'plamlaridan o'tadi. Shuning uchun dasturda P massiv kiritilgan. Uning elementlari bilan « b » sonining ikkilik razryadlari kabi ish olib boriladi.

Dastur

Program Ichki_to'plam_summasi;

const nn=100;

label l;

var

m,n,i,s :integer;

a : array [1..nn] of integer;

b : array [1..nn] of boolean;

begin

writeln ('n,m=');

readln (n,m);

for i:=1 to n do

begin

```

        writeln ('a[',i,']=');
        readln (a[i]);
        b[i]:=false;
    end;
s:=0;
repeat
    for i:=1 to n do if (b[i]) then
        begin
            b[i]:=false; s:=s-a[i]
        end
        else
            goto 1;
1: b[i]:=true; s:=s+a[i];
until s=m;
for i:=1 to n do
    if (b[i]) then write(1:4);
    writeln;
end.

```



229-masala
Algoritm. «Labirint»

Yaxshi ma'lum bo'lgan bu masala, bu yerda biroz o'zgartirilgan. Yechim ikki qismga bo'linadi: chiqish yo'lini izlash va «teskari yo'l»ni (yo'lovchining chiqishdan boshlab, boshlang'ich holatigacha) chop etishdan iborat.

Birinchi qismning oddiy yechimi quyidagicha bayon etiladi: Yo'lovchining boshlang'ich nuqtasi bo'lgan $a[i,j]$ katakka 2 soni yoziladi va $k=2$, deb olinadi. A labirintning hamma kataklari ko'zdan kechirib chiqiladi. Yozilgan har bir katak uchun uning to'rtta qo'shnisi ko'zdan kechiriladi. Agar unga qo'shni bo'lgan to'rtta katakdan, hech bo'lmaganda, bittasida k (hozir 2 ga teng) yozilgan bo'lsa, ko'rilyotgan a katakka $k+1$ soni yoziladi. Endi $k=k+1$ oshiriladi va A ning hamma kataklari qaytadan ko'zdan kechiriladi. Agar son chegara katakka yozilgan bo'lsa, (yo'l topilgan) yoki A ning hamma kataklarini ko'zdan kechirishda, u bironta ham kattakka yozilmagan bo'lsa (yo'l yo'q), jarayon tugaydi. Qisqa yo'lda nechta katak bo'lsa, massivning hammasini qarab chiqish ham shuncha bo'ladi.

Oldingi algoritmni yaxshilash oson. Avvalgidek, har qadamda A labirintning hamma kataklari ko'zdan kechiriladi. Agar katakda,

nol yozilgan bo'lsa-yu, qandaydir unga qo'shni katakda $K \geq 2$ son bo'lsa, a katakka $k+1$ soni yoziladi. Bu masala birinchi qismining yechimi ekanligi aniq. Lekin, agar omad chopsa, bu yerda yechim tez topiladi.

Boshlang'ich katakdan boshlab (boshda 2 yozilgan katak), birinchi bo'sh (ya'ni, 0 lik katak) qo'shnini izlovchi dastur samarali bo'ladi. Qo'shni bo'sh katakka 3 soni yoziladi va 4 sonini yozish uchun uning bo'sh qo'shnisi izlanadi va hokazo. Chegaraga chiqilganda (yo'l topildi) yoki bo'sh qo'shnilar bo'lmaganda berk katakda jarayon to'xtatiladi. Agar berk katak boshlang'ich katakda paydo bo'lsa, chiqish yo'q. Agar berk katak boshqa bo'lib, unga $k > 2$ soni yozilgan bo'lsa, unga bir sonini yozish (uni yopish) va qo'shni, $k-1$ sonli katakka o'tish kerak. Bunday katak bor va u yagona. Bu yechimda variantlarni ko'rib chiqishning umumiy chizmasini topish oson. Labirint to'g'risidagi masala yechimini rekursiv yozib, sezilarli qisqartirish mumkin. Bu ish taklif etilgan dasturda bajarilgan. Lekin bunday dasturni o'qishdan ko'ra yozish oson.

Keltirilgan algoritmlar (birinchisidan boshqa) eng qisqa yo'lni bermasligi ham mumkin. Labirintda qisqa yo'lni tejamli izlash uchun ko'zdan kechirilishi kerak bo'lgan katak (x, y) koordinatalari ro'yxati uchun maxsus X va Y massivlarni tashkil etish mumkin. Bunday usul **kengligi (eni) bo'yicha izlash** deyiladi.

Avval X, Y ga yo'lovchining boshlang'ich katak koordinatalari kiritiladi. Har qadamda X va Y massivlardan, ko'zdan kechirish uchun navbatdagi katak («b» raqamli) koordinatalari olinadi, uning bo'sh qo'shnilari esa ixtiyoriy tartibda X, Y ro'yxat davomida «e» raqam bilan yozib qo'yiladi. Shunday qilib, ro'yxat boshidan ishlab chiqiladi va oxiridan uzaytiriladi.

Izlash labirint chegarasida bo'sh katakka yotish bilan yoki X, Y ro'yxatni tugatish bilan (chiqish bo'lmasa) tamomlanadi.

Teskari yo'lni izlash va chop etish rekursivmas dasturlarda bir xil, rekursivlarda esa avtomatik bajariladi.

Dastur

Program Labirint;

const mm=15; nn=15;

var

i, n, j, m : integer;

chik : boolean;

a : array [1..mm, 1..nn] of byte;

```

procedure L (i,j:integer);
begin
  if not chik then
    if a[i,j]=0 then
      begin
        if i=1 or i=m or j=1 or j=n then chik:=true;
        a[i,j]=1; L(i,j-1); L(i, j+1); L(i-1,j); L(i+1,j);
        if chik then writeln (i, ' ',j);
        end;
      end;
begin
  writeln ('m,n=');
  readln (m, n);
  for i:=1 to m do
    for j:=1 to n do
      begin
        writeln ('a[',i,',',j,']=');
        readln (a[i, j]);
      end;
    writeln;
    writeln ('i, j=');
    readln (i, j)
    writeln;
    chik:=false;
    if not chik then writeln ('chiqish yo`q');
  end.

```



230-masala

Algoritm. «Mukammal sonlar»

Natural i son mukammal bo'lishini hal qilish uchun hamma $j = 1, 2, \dots, i-1$ sonlarni ko'rib chiqish, ularning qaysi biri i sonining bo'luvchisi bo'lishini aniqlash va bu bo'luvchilarni qo'shish kerak.

Bo'luvchiga «nomzodlar»ni $i-1$ gacha emas, balki \sqrt{i} gacha ko'rib chiqib, dastur bajarilishini tezlashtirish mumkin. Buning uchun $k = i / j$ ($k \geq j$) ni topish va agar i sonining bo'luvchisi j bo'lsa, faqat j ni emas, balki k ni ham hisobga olish kerak. Faqat bironta ham bo'luvchi ikki marta olinishiga ($j=k$ da) yo'l qo'ymaslik kerak. Bu algoritm taklif qilingan dasturda keltirilgan.

Dastur

Program Mukammal_sonlar;
var

m, i, j, k, s :integer;

begin

write ('m:=');

readln (m);

writeln ('program get m=', m);

for i:=2 to m do

begin

S:=1; j:=1;

repeat

J:=J+1;

*K:=i*div j;*

*if (i=k*j) and (j<=k) then*

begin

s:=s+j if j>k then s:=s+k;

end;

until j>=k

if s=i then writeln (i);

end;

end.



231-masala

Algoritm. «Hosil qilinmaydigan son»

Bu masala birinchi qarashdagidan ko'ra qiziqroq, chunki u $n*n$ (hatto $n*\log n$) amalda yechiladi. Mumkin bo'lgan yechim quyidagicha $S=1$ deb olinadi, P massivda qandaydir $P(j) \leq S$ element borligi aniqlanadi. Agar bunday element bo'lmasa, «S» yechim bo'ladi. Agar bunday element topilsa, uning qiymati S ga qo'shiladi, ya'ni $S=S+P[j]$ Massivdan $P(j)$ olib tashlanadi va yana qolgan elementlar ichidan «S» ning yangi qiymatidan katta element borligi izlanadi va hokazo. Induksiya usuli bo'yicha xuddi shunday tarzda birinchi tashkil etilmaydigan sonni topishni isbotlash mumkin. Haqiqatda navbatdagi $i=1,2,\dots$, qadamda P massiv elementlarini $P(i)$ dan boshlab ko'rib chiqish, topilgan $P(j)$ elementni esa $P(i)$ elementga almashtirish kerak.

Harakatlar (amallar) sonini tejash maqsadida (massivda sonlar eng yomon joylashgan holda) oldindan P massivning faqat navbatdagi elementini taqqoslash kerak bo'ladi. Bayon etilgan algoritm taklif qilingan dasturda amalga oshirilgan.

Dastur

```
Program Hosil_qilinmaydigan_son;  
const NN=10;  
label 1,2;  
var  
    n,i,j,s : integer;  
    P : array [1..NN] of integer;  
begin  
    writeln ('n:=');  
    readln (n);  
    for i:=1 to n do readln (P[i]);  
    writeln;  
    S:=1;  
    for i:=1 to n do  
        begin  
            for j:=i to n do if P[j]<=s then goto P1;  
            goto 2;  
            1: s:=s+P[j]; P[j]:=P[i];  
            end;  
            2 : writeln(s);  
        end.  
end.
```



232-masala Algoritm. «Do'mbira»

Quyidagi oddiy yechim taklif etiladi. K — «kichiklikka nomzod» vektor raqami, p — navbatdagi vektor raqami bo'lsin. Boshida $k=1$ va $p=2$, deb olinadi.

Agar qandaydir bir lahzada quyidagilar aniqlansa:

- 1) $X_k \leq X_p$ bo'lsa, p birga oshiriladi;
- 2) $X_k > X_p$ bo'lsa, k p ga almashtiriladi va p birga oshiriladi.

Agar shundan keyin $p \leq 12$ bo'lsa, X_k va X_p vektorlar yangi juftini taqqoslashga o'tish kerak. Agar $p > 12$ bo'lsa, sonlar yechimi bo'ladi. Bu algoritm taklif etilgan dasturda amalga oshirilgan.

Boshqa xil yechim ham bor. U ba'zi bir hollarda tezroq hosil qilinishi mumkin. Unga ko'ra «kichiklikka nomzod» vektorlar ro'yxati uchun qo'shimcha $L[1:12]$ massiv tashkil qilinadi (aniqrog'i «nomzod»lar boshlanadigan raqamlar ro'yxati uchun). Boshda bu 1 dan 12 gacha raqamlar bo'ladi. Shundan so'ng ularning ichidan birinchi elementi — kichik bo'lganlari qoldiriladi, ulardan ikkinchi

elementi — kichik bo‘lganlari va hokazo bironta ham nomzod qolmaguncha yoki minimal vektorlarga nomzod hamma 12 ta element qarab chiqilmaguncha davom ettiriladi.

Dastur

```

Program Do‘mbira;
const N=12;
label 1;
var
    i, j, p, l, k :integer;
    A : array [1..n] of real;
begin
    for i:=1 to n do
        begin
            writeln ('A[',i,']:=');
            readln (A[i]);
        end;
        k:=1;
        for P:=2 to n do
            begin
                for i:=0 to n-1 do
                    begin
                        j:=k+i; if j>n then j:=j-n;
                        l:=P+j; if l>n then l:=l-n;
                        if A[l]<A[j] then k:=p;
                        if A[l]>A[j] then goto 1;
                    end;
                l: end;
            writeln;
            writeln (k);
        end.

```



233-masala Algoritm. «Kublar yig‘indisi»

$M=0$ ni olamiz $i-1$, j esa birinchi son, uning uchun $j^3+1 \geq N$. $K=i^3+j^3$ deb olamiz va « K » ni « N » bilan taqqoslaymiz. Uch hol bo‘lishi mumkin:

- (1) agar $k < N$ bo‘lsa, $i=i+1$ ni hosil qilamiz.
- (2) agar $K > N$ bo‘lsa, $j=j-1$ ni hosil qilamiz.
- (3) agar $K = N$ bo‘lsa, $M=M+1; j=j-1; i=i+1$ ni hosil qilamiz.

Bu hisoblashlar $i \leq j$ bo'lguncha davom ettiriladi. Shuningdek, (3) holda $i=i+1$ o'rninga darrov $i=i+2$ yozish qulayligini bildiramiz.

Boshida $i \leq j \leq \sqrt[3]{N} + 1$ bo'lgani va har bir qadamda i o'sgani yoki j kamaygani uchun qadamlar $\sqrt[3]{N}$ dan katta bo'lmaydi.

Algoritm to'g'riligini isbotlash uchun $P(i,j)=\{(x,y)\}$ orqali $i \leq x \leq y \leq j$ va $x^3+y^3=N$ shartlarni qanoatlantiruvchi shunday x va y natural son juftlari to'plamini belgilaymiz.

Faraz qilamiz, $k=i^3+j^3$ bo'lsin, agar $k < N$ bo'lsa, $P(i,j)$ to'plamdagi ixtiyoriy (x,y) juftlik uchun $i^3+j^3 < x^3+y^3$ bo'ladi, lekin $y \leq j$, bundan $i < x$ va, demak, agar $k < N$ bo'lsa, $P(i,j)=P(i+1,j)$.

Xuddi shunga o'xshash, agar $k > N$ bo'lsa, $P(i,j)=P(i,j-1)$, nihoyat, $k=N$ da, (i,j) juftlik $P(i,j)$ dan olib tashlanganidan keyin, agar $k=N$ bo'lsa, $P(i,j) \setminus (i,j) = P(i+1, j-1)$ bo'ladi.

Dastur

```

Program Kublar_yig'indisi;
var
    i, j, m, k, n    : integer;
begin
    writeln ('N:= ');
    readln (n);
    m:=0; j:=1; i:=1;
    while j*j*j+1 < n do j:=j+1;
    repeat
        k:=i*i*i+j*j*j;
        if k=n then m:=m+1;
        if k <= n then i:=i+1;
        if k >= n then j:=j-1;
    until i > j;
    writeln (m);
end.

```



234-masala

Algoritm. «Oddiy bo'luvchilar».

$i=2,3,\dots$ sonlarini i soni N sonining bo'luvchisi bo'lguncha almashtiramiz. N ni i ga bo'lamiz,

$$N = N / i$$

va bo'lishni N ni i ga bo'linguncha takrorlaymiz. Shundan keyin, $N > 1$ bo'lsa, i ning navbatdagi qiymatiga o'tamiz va hokazo. Shunday

tarzda boshlang'ich N sonining hamma oddiy bo'luvchilarini va faqat ularni topish mumkinligini sezish qiyin emas.

Boshda 0 ga, keyinchalik, oxirgi bo'luvchiga teng bo'lgan, yana bitta j o'zgaruvchini kiritamiz. Yangi bo'luvchi birinchi marta j dan farq qiladi va chop etiladi. Dastur ishini tezlashtirish uchun, $i=2$ holini alohida ko'rish va keyinchalik esa i ning toq qiymatlari bo'yicha siljish mumkin.

Dastur

Program Oddiy_bo'luvchilar;

var

i,j,n : integer;

begin

writeln ('N:= ');

readln (n);

writeln ('program get N:= ');

j:=0; i:=2;

while n>1 do

begin

while (n mod i)<>0 do i:=i+1;

if i<>j then

begin

writeln(i);

j:=i;

end;

n:=n div i;

end;

end.



235-masala

Algoritm. «Ko'phad»

$P_0(x)=1$ deb olib, $P_m(x)=a(m,0)+a(m,1)x+\dots+a(m,m-1)x^{m-1}+x^m$ ko'phadni aniqlovchi, dastlabki m ta qavs ko'paytuvchilarini $P_m(x)$ deb belgilab olish qulay:

$$P_m(x) = (x-x[1]) * \dots * (x-x[m]).$$

$P_m(x) = P_{m-1}(x)(x-x[m])$ ekanligi kelib chiqadi va $a(m,i)$ koeffitsientlar $a(m-1,i)$ orqali quyidagi tartibda ifodalanadi:

$$a(m,m-1) = a(m-1,m-2)-x[m]$$

$$a(m,i) = a(m-1,i-1)- a(m-1,i)x[m]$$

$$a(m,0) = - a(m-1,0)x[m].$$

Agar $a(m,i)$ koeffitsient A massivning $a[i]$ elementiga joylashtirilsa, unda ko'phadning P_{m-1} koeffitsientlaridan, P_m ko'p-

hadning koeffitsientlariga o'tish ($m > 1$ da) quyidagi operatorlar orqali bajariladi:

$$a[m-1] = a[m-2] - x[m]$$

$$a[i] = a[i-1] - a[i]x[m], \quad i = m-2, \dots, 1 \text{ uchun}$$

$$a[0] = -a[0] - [m]$$

Bu ish taklif qilingan dasturda amalga oshirilgan. U boshlang'ich va so'nggi qiymatlarga tuzatishlar bilan ikkilamchi tashqi (m bo'yicha 1 dan n gacha) va ichki (i bo'yicha $m-1$ dan 0 gacha) siklga ega.

Dasturni boshqacha qilib ham, tashqi siklni i bo'yicha $n-1$ dan 0 gacha va ichkisini $n-i$ dan n gacha, rasmiylashtirish mumkin edi.

Dastur

Program Ko'phad;

const NN=100;

var

i, m, n : integer;

x : array [1..NN] of real;

A : array [1..NN-1] of real;

begin

Writeln ('N:=');

readln (n);

for i:=1 to n do

begin

writeln ('x['i,']:=');

readln (x[i]);

end;

a[0]:=-x[1]; a[1]:=1;

for m:=2 to n do

begin

a[m-1]:=a[m-2]-x[m];

i:=m-2;

while i>=1 do

begin

*a[i]:=a[i-1]-a[i]*x[m];*

i:=i-1;

end;

*a[0]:=-a[0]*x[m];*

end;

writeln;

for i:=0 to n-1 do writeln ('a['i,']:=',a[i]);

end.



236-masala
Algoritm. «Qisqa ko'paytuvchilar»

$A[n]$ massivda «qisqa ko'paytuvchi»larni hosil qilib va o'sib borish tartibida joylashtirib boramiz. Dastlabki birinchi i qisqa ko'paytuvchilar hosil qilingan bo'lsin:

$$a[1] < \dots < a[i] (*).$$

Va (*) ga kiruvchi sonlarning k_2 va k_3 (ular uchun $a_2 = 2 * a[k_2] + 1$ va $a_3 = 3 * a[k_3] + 1$ tengliklar bajariladi) minimal raqamlari ma'lum bo'lsin, k_2, k_3 lar (*) ga kirmaydi.

(*) ga yana bitta had kiritilganda nima bo'lishini ko'ramiz. Ma'lumki,

$$a[i+1] = \min(a_2, a_3).$$

Agar $a[i+1] = a_2$ bo'lsa, k_2 1 ga, agar $a[i+1] = a_3$ bo'lsa, k_3 1 ga (agar $a[i+1] = a_2 = a_3$ bo'lsa, ikkala k_2 va k_3 sonlar 1 ga) o'sadi. Avval $i=1$ $a[1]=1$, $k_2=1$, $k_3=1$. Navbatdagi $a[2]$, $a[3]$, ... hadlar bayon etilgan algoritm bilan hisoblanadi.

Dastur

Program Qisqa_ko'paytuvchilar;

const NN=100;

var

i, k_2, k_3, a_2, a_3, n : integer;

A : array [1..NN] of integer;

begin

writeln ('N:=');

readln (n);

$k_2:=1$; $k_3:=1$; $a[1]:=1$;

writeln;

for $i:=2$ to n do

begin

$a_2:=2*a[k_2]+1$;

$a_3:=3*a[k_3]+1$;

if $a_2 \leq a_3$ then

begin

$a[i]:=a_2$;

$k_2:=k_2+1$;

end;

if $a_3 \leq a_2$ then

```

begin
    a[i]:=a3;
    k3:=k3+1;
end;
writeln ('a[i]:=');
end;
end.

```



237-masala Algoritm. «Poker»

Masalaning yechimi oddiy. Berilgan A massivda nechta teng juftliklar borligi sanaladi. Ularning soni kombinatsiyani bir qiymatli tavsiflar ekan. Teng juftlar sonini hisoblash uchun S hisobchini nolga tenglashtiramiz, $a[i]$ ning har bir elementini $a[i]$ ning keyingi qolganlari bilan $j=i+1, i+2\dots$ uchun taqqoslaymiz va $a[i]=a[j]$ bo'lgan holda, S ga 1 tadan qo'shamiz.

Chop etish uchun kerakli P javob (birinchi ikkita holdan boshqasida, $P=7-S$) S hisobchi (schyotchik)ning qiymati bo'yicha jadvalda hosil bo'ladi:

	S	P	S	P
$4+3+2+1$	$=10$	1	$1+1 = 2$	5
$3+2+1$	$=6$	2	$1 = 1$	6
$2+1+1$	$=4$	3	$0 = 0$	7
$2+1$	$=3$	4		

Dastur

Program Poker;

var

i, j, p :integer;

A : array [1..5] of integer;

begin

p:=0;

for i:=1 to 5 do readln (A[i]);

writeln;

p:=0;

for i:=1 to 4 do

for j:=i+1 to 5 do

if A[i]=A[j] then p:=p+1;

```

if P=10 then writeln (1)
else
if P=6 then writeln (2) else writeln (7-P);

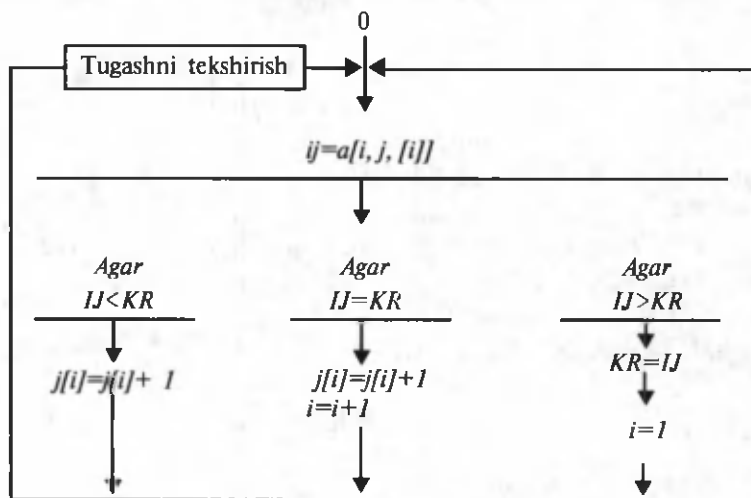
```

end.



238-masala
 Algoritm. «Teng elementlar»

Har bir i satr uchun $J[i]$ da, bu satrda borib yetilgan ustun raqamini eslab qolish maqsadida, $J[1:m]$ massiv tashkil qilish va boshda ham $j[i]=1$ deb olish kerak. Teng elementlarga nomzodlarni KR deb, A massivning navbatdagi ko'riladigan elementini esa IJ deb belgilaymiz. Avval, yana $KR=a[1,1]$ va $i=1$ deb olamiz. Keyingi (kamdan-kam uchraydigan) holni blok-chizma yordamida bayon etish mumkin:



$IJ=A[i, j[i]]$ o'zlashtirishdan oldin tugatishni tekshirishni qo'shib qo'yamiz. Agar $i > m$ bo'lsa, kerakli qiymat topilgan. Agar $j(i) > n$ bo'lsa, bu qiymat yo'q.

Lekin yordamchi $j[1:n]$ ustun o'rniga yordamchi $io[1:n]$ satrni tashkil qilish mumkin. Avval unga A massivning birinchi satr kiritiladi, io satrning no «harakatdagi» elementlar soni « n »ga teng deb olinadi. Keyin massivning navbatdagi $i=1, 2, 3, \dots, m$ satr io satr bilan taqqoslanadi. io satrda faqat i satrda uchragan elementlar qoldiriladi (va boshiga suriladi), io da qolgan no elementlar soni esa yangi qiymat qabul qiladi. Agar no qandaydir bir qadamda nolga aylansa, teng elementlar yo'q, aks holda izlanayotgan element $io[1]$ bo'ladi. Bu algoritm taklif etilgan dasturda amalga oshirilgan. Dasturda $io[j]$ elementlar o'rniga $a[o, j]$ elementlar olingan.

Dastur

```
Program Teng_elementlar;
const NN=20;
      MM=20
label 1,2;
var
    m,n,i,j,jo,no,jn : integer;
    A : array [1..MM,1..NN] of integer;
begin
    writeln ('m,n:=');
    readln (m,n);
    for i:=1 to m do
        for j:=1 to n do
            begin
                writeln ('A['i','o'j,']:=');
                readln (A[i,j]);
            end;
        for i:=1 to n do A[o,j]:=A[1,j];
        no:=n;
        for i:=2 to m do
            begin
                j:=1; jo:=1; jn:=1;
                1: if A[o,jo]<A[i,j] then jo:=jo+1
                    else
                        if (a[o,jo]>a[i,j]) then j:=i+1
                            else
                                begin
                                    a[o,jn]:=a[o,jo];
                                    j:=j+1;
                                    jn:=jn+1;
                                    jo:=jo+1;
                                end;
                                if (jo<no) and (j<=n) then goto 1;
                                no:=jn-1;
                                if j<=n then goto 1;
                                if no=0 then goto 2;
                            end;
                writeln (a[o,1]); exit;
                2: writeln ('Teng elementlar yo'q');
            end.
end.
```



239-masala
 Algoritm. «Yo'l»

i_1, i_2, \dots, i_n sonlarning hamma variantlarini ko'rib chiqish mumkin emas, ularning soni m^n ta. Lekin masalani yechishning boshqa, $k=1, 2, \dots, n-1$ bo'yicha induksiya yo'li mavjud. k va i_{k+1} ning belgilangan sonlarida $B[k, i_{k+1}] = \min(A[i_1, i_2, 1] + \dots + A[i_k, i_{k+1}, k])$ ni qo'yamiz, bu yerda \min mumkin bo'lgan i_1, i_2, \dots, i_k to'plamlardan olinadi. Unda:

$B[1, i_2] = \min(A[i_1, i_2, 1])$ hamma i_1 lar bo'yicha,

$B[2, i_3] = \min(B[1, i_2] + A[i_2, i_3, 2])$ hamma i_2 lar bo'yicha,

$B[n-1, i_n] = \min(B[n-2, i_{n-1}] + A[i_{n-1}, i_n, n-1])$ hamma i_{n-1} lar bo'yicha va izlanayotgan $R = \min B[n-1, i_n]$ hamma i_n lar bo'yicha ekanligini ko'rish oson. Shunday qilib, bitta $B[k, i_{k+1}]$ ni hisoblash uchun faqat m ta variant (i_k tanlov) qaraladi. Belgilangan k va hamma $B[k, i_{k+1}]$ lar uchun m^2 variantlar ko'rib chiqiladi, masalaning hammasi esa $m^2 \cdot n$ dan kam bo'lmagan variantlarni ko'rib chiqishni talab qiladi.

Dastur

Program Yo'l;

const mm=10; nn=10;

var

k, i, n, j, m : integer;

x, r : real;

A : array [1..mm, 1..mm, 1..nn] of real;

B, C : array [1..mm] of real;

begin

writeln ('m, n=');

readln (m, n);

for i:=1 to m do

begin

b[i]:=0;

for j:=1 to m do

for k:=1 to n-1 do

begin

writeln ('a[', i, ', ', j, ', ', k, ']=');

readln (a[i, j, k]);

end;

end;

for k:=1 to n-1 do

```

begin
  for j:=1 to m do
    begin
      r:=b[i]+ a[i,j,k];
      for i:=2 to m do
        begin
          x:=b[i] + a[i,j,k];
          if x<r then r:=x;
        end;
      c[j]:= r;
    end;
  for j:=1 to m do b[j]:=c[j];
end;
r:=b[1]; for i:=2 to m do
  if b[i] < r then r:=b[i];
writeln (r);
end.

```



240-masala

Algoritm. «Uchburchak va nuqta»

Quyidagini aytib o'tamiz. $M(x,y)$ nuqta $M1(x1,y1)$ va $M(x2,y2)$ nuqtadan o'tuvchi $L12$ to'g'ri chiziqda yotsin. Unda uchburchaklarning o'xshashligidan quyidagini hosil qilamiz:

$$\frac{y - y1}{x - x1} = \frac{y2 - y1}{x2 - x1}$$

Maxrajlarning nolga aylanishi bilan bog'liq bo'lgan «qopqon»ni chetlab o'tib, bu tenglamani quyidagi ko'rinishda yozamiz:

$$F_{12}(x,y) \equiv (x-x1) \cdot (y2-y1) - (x2-x1) \cdot (y-y1) = 0.$$

$L12$ to'g'ri chiziqda yotmagan $M(x,y)$ nuqtalar uchun $F_{12}(x,y)$ funksiyaning nolga aylanmasligi aniq. Bundan tashqari $L12$ to'g'ri chiziqdan chap tomonda yotgan (agar $M1$ dan $M2$ yo'nalishda qaralsa) M nuqtalar uchun $F_{12}(x,y) < 0$ va $L12$ dan o'ng tomondagi M nuqtalar uchun $F_{12}(x,y) > 0$ ekanligini tekshirish qiyin emas. Lekin $L12 = (M1 - (-M2))$, $L23 = (M2 - (-M3))$ va $L31 = (M3 - (-M1))$ to'g'ri chiziqlar uchun hosil qilingan uchta sonning: $F_{12}(x,y)$, $F_{23}(x,y)$, $F_{31}(x,y)$ hammasi bir xil ishoraga ega bo'lsa, M nuqta $(M1, M2, M3)$ uchburchak ichida shu vaqtda va faqat shu vaqtda yotishini bildiradi. Endi masalani yechish uchun $M1$, $M2$, $M3$

nuqtalarning koordinatasi uchun $x[1:3]$ va $y[1:3]$ ikkita massivni kiritish va $J(i,j) = \text{sign}[(X-X[i])*(Y[j]-Y[i])-(X[j]-X[i])*(Y-Y[i])]$ funksiyani aniqlash qulay.

Unga uch marta murojaat etib, quyidagi sonlarni hosil qilamiz:
 $t_1 = F(1,2)$, $t_2 = F(2,3)$, $t_3 = F(3,1)$.

Agar ular teng bo'lsa, M nuqta uchburchak ichida, teng bo'lmasa, uchburchak tashqarisida (yoki chegarada) yotadi.

Agar dasturlash tili ham, massiv ham o'zgaruvchini bitta harf bilan belgilashga imkon bermasa, M nuqta koordinatalarini $x[0]$, $y[0]$ yoki xt , yt va h.k. shunga o'xshab belgilash mumkin.

Bu masalaning boshqa yechimi yuzalarni taqqoslashga asoslangan. Berilgan ($M1$, $M2$, $M3$) uchburchak yuzi S bilan, M nuqtani $M1$, $M2$, $M3$ uchlarning ikkitasi bilan tutashtirganda hosil bo'ladigan uchburchaklar yuzalari esa $S1$, $S2$ va $S3$ bilan belgilanadi. Agar $S=S1+S2+S3$ bo'lsa, nuqta uchburchak ichkarisida, aks holda tashqarida yotgan bo'ladi. Bu yechim taklif etilgan dasturda keltirilgan.

Uchburchak yuzi Geron formulasi bilan hisoblanadi. Agar $S1+S2+S3 > 1,000001 * S$ bo'lsa, nuqta uchburchak tashqarisida yotgan deb hisoblanadi. Hisoblash xatoligini $1,000001$ ko'paytiruvchi hisobga oladi.

Dastur

```
Program Uchburchak_va_nuqta;  
type Point= record;  
    x : real;  
    y : real;  
end;  
var  
    TR : array [1..3] of Point;  
    t : Point;  
    i : 1..3;  
function Dis(p,g : Point) : real;  
begin  
    Dis:=sqrt(sqr(p.x-g.x)+sqr(p.y-g.y));  
end;  
function Grn(a,b,c:Point):real;  
var da,db,dc,p:real;  
begin  
    da:=Dis(c,b); db:=Dis(a,c); dc:=Dis(a,b);  
    p:=(da+db+dc)/2;  
    Grn:=sqrt(p*(p-da)*(p-db)*(p-dc));
```

```

end;
begin
  for i:=1 to 3 do
    begin
      writeln ('x',i,',',y',i,',');
      readln (tr[i].x,tr[i].y);
    end;
  writeln ('x,y=');
  readln (t.x,t.y);
  if Grn (tr[1],tr[2],tr[3])*1.000001< Grn(t,tr[1],tr[2]+
  Grn(t,tr[1],tr[3])+
  Grn(t,tr[2],tr[3])) then writeln('yo'q')
    else writeln ('ha');
end.

```



241-masala Algoritm. «Tartiblangan kasrlar»

Dasturni yordamchi massivlarsiz va kasrlar qisqarishini tekshir-
masdan tuzish mumkin. Buning uchun m/n kasr kiritiladi va avval
 $m=0$, $n=1$ deb hisoblanadi, m/n chop etiladi. Keyin hamma a/b
kasrlar ichidan m/n dan kattalari va $b \leq p$ ($p=7$ shartda)lardan eng
kichigi tanlanadi. Bu i/j bo'lsin. Agar $i/j < 1$ bo'lsa, m/n kasr i/j
kasr bilan almashtiriladi va jarayon davom ettiriladi.

Dasturning ba'zi bir xususiyatlarini ko'rsatib o'tamiz.

Berilgan $b=2, \dots, p$ maxrajlar uchun surat tanlanmaydi, balki
 $a=m*b/n+1$ formula hisoblanadi.

Hamma m/n ga teng bo'lgan kasrlar ichidan, u eng kichik max-
rajga ega ekanligidan, har bir topilgan m/n kasr avtomatik ravishda
qisqarmaydigan bo'ladi. Shuningdek, kasrlarni taqqoslashda xusu-
siylarini (mashinada taqriban ifodalanadiganlarini) emas, balki
ko'paytmalarini ($a/b < i/j$ emas-u, lekin $a*j < b*i$ bo'lganligini)
taqqoslash yaxshiligini ta'kidlaymiz.

Dastur

Program Tartiblangan_kasrlar;

var

p, m, n, a, b, i, j :integer;

begin

writeln ('p=');

readln (p);

m:=0; n:=1;

```

repeat
  writeln (m:4,'/',n:4,'=',m/n);
  i:=1; j:=1;
  for b:=2 to p do
    begin
      a:=m*b div n+1;
      if a*j<b*i then
        begin
          i:=a; j:=b;
        end;
    end;
  m:=i; n:=j;
until i>=j;
end.

```



242-masala Algoritm. «To'rtliklar jufti»

Bu masala katta bo'lmagan «qoidaga» ega, chunki kerakli xossali minimal sonlar o'rniga, qo'shiluvchilarni ko'rib chiqishning ma'lum bir tashkil etilishida uchrashi mumkin bo'lgan birinchi sonni hosil qilish mumkin. Izlanayotgan sonning ikki xil ifodalanishi (kvadratlar yig'indisi ko'rinishida) eng katta qo'shiluvchi bilan farqlanishi kerakligi hisobga olinsa, dastur ishini yana ham tezlashtirishi mumkin.

Dastur

```

Program To'rtliklar_jufti;
label 1,2;
var
  i,n,j,k,p : integer;
  i1,j1,k1,p1 :integer;
  b : boolean;
begin
  for n:=2 to max int do
    begin
      b:=false;
      for i:=1 to n trunc (sqrt(n)) do
        begin
          for j:=1 to i do
            begin
              if i*i+j*j>=n then goto 1;
            end;
          end;
        end;
      b:=true;
    end;
  end;
  goto 2;
end.

```

```

                                for k:=1 to j do
                                    if i*i+j+k*k<n then
begin
    p:=trunc(sqrt(n-i*i-j*j-k*k));
    if (i*i+j*j+k*k+p*p=n) and (p<=k) then
        begin
            if (b) then goto 2;
            i1:=i; j1:=j; k1:=k; p1:=p;
            b:=true;
        end;
    end;
end;
end;
1: end;
end;
2: writeln (i*i+j*j+k*k+p*p, '=');
writeln (i, '*', i, '+', j, '*', j, '+', k, '*', k, '+', p, '*', p, '=');
writeln (i1, '*', i1, '+', j1, '*', j1, '+', k1, '*', k1, '+', p1, '*', p1, '=');
end.

```



243-masala Algoritm. «M+1 sonni ikkilikda yozish»

a_0, a_p, \dots , sonlar ko'zdan kechiriladi. Birinchi uchragan nolgacha birlar nollar bilan almashtirib boriladi. Birinchi nol uchragach, almashtirish to'xtatiladi. Faqat, yechim shartdagi kabi «n» ta emas, balki $n+1$ ta songa ega bo'lishi mumkinligini hisobga olish kerak.

Dastur

Program Sonni_ikkilikda_yozish;

var

i, j, n : integer;

b : boolean;

begin

writeln ('n:=', n);

readln (n);

b:=true;

for i:=1 to n do

begin

readln(j);

if b then writeln (' ', 1-j) else

```

                                writeln (' j');
                                if j:=0 then b:=false;
                                end;
                                if b then writeln ('1');
end.

```



244-masala
 Algoritm. «Kvadratlar yig'indisi»

Dastur oson o'qiladi.

Bu masala «kvadrat ildiz» funksiyasi qo'llanilmaganida ham yechimga ega. Uning yechilishini, tekshirilayotgan elementni oldingisi bilan emas, keyingisi bilan, birinchi takrorlashgacha taqqoslab, tezlashtirish mumkin. Bunday taqqoslashlar soni $m * m$ dan $m * k$ gacha kamayadi, bu yerda m -hamma sonlar miqdori, k -berilgan massivdagi har xil sonlar miqdori.

Dastur

Program Kvadrat_summasi;

var

i,j,m : integer;
ff : boolean;

begin

readln (m);
write('m:=',m);
i:=1;
ff:=true;
*while (2*i*i<=m) and ff do*

begin

*j:=round (sqrt(m-i*i));*
*if (i*i+j*j=m) then ff:=false*
else i:=i+1

end;

if ff then writeln('yo'q')
else writeln(i,'x',i,'+',j,'x',j,'=',m);

end.



245-masala
 Algoritm. «O'rin almashtirish»

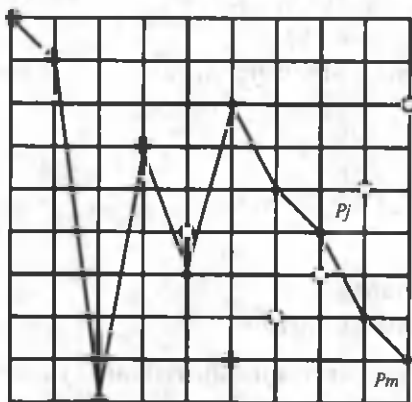
Bu masalaning xususiyati hamma o'rin almashtirishlarni hosil qilishdadir. Bu bilan u umumiy tanlash masalasidan farq qiladi va masalaning yechilishini yengillashtiradi. Shunga qaramay, masalaning bayoni dasturdan uzunroqdir.

1 dan « m » gacha bo'lgan sonlarning barcha o'rin almashtirishlarini ko'z oldimizga keltiramiz. Xayolan, ularni lug'at tartibida joylashtiramiz va amalda esa berilgan o'rin almashtirish bo'yicha bevosita quyidagini hosil qilishni o'rganamiz. Shunda birinchi o'rin almashtirishdan $(1, 2, \dots, m)$, biz ketma-ket bu sonlarning barcha o'rin almashtirishlarini hosil qilamiz. Yo'l-yo'lakay $1, 2, \dots, m$ sonlarning har bir $P=(P_1, P_2, \dots, P_m)$ o'rin almashtirishi uchun berilgan A sonlarning xuddi shunday $(A(P_1), A(P_2), \dots, A(P_m))$ o'rin almashtirishini chop etamiz.

$1, 2, \dots, m$ sonlarning berilgan $P=(P_1, P_2, \dots, P_m)$ o'rin almashtirishi bo'yicha, bevosita navbatdgisini hosil qilish uchun, P_1, P_2, \dots, P_m sonlar oxiridan qarab chiqish kerak. O'ng tomondagi hadidan kichik bo'lgan ($P_i < P_{i+1}$), birinchi uchragan P_i hadda to'xtalinadi. Agar bunday had bo'lmasa, P o'rin almashtirish $(m, m-1, \dots, 1)$ ko'rinishga ega (ya'ni oxirigisi) bo'ladi. $P_{i+1} > P_{i+2} > \dots > P_m$ hadlar kamayuvchi ketma-ketlikni hosil qilishi aniq. Ularning ichidan P_i dan katta bo'lgan (agar oxiridan qaralsa) birinchi P_j hadni topamiz va ularning o'rnini almashtiramiz.

$P_{i+1}, P_{i+2}, \dots, P_m$ hadlarni o'sib borish tartibida almashtirib qo'yish qoladi va izlangan (uni $Q=(g_1, \dots, g_m)$ deb ataymiz) o'rin almashtirish hosil bo'ladi (4-rasm).

Haqiqatan ham, $P > Q$, ularning birinchi $i-1$ hadlari mos tushadi, P_i esa g dan kichik ($P_i < g_i$) bo'ladi (chunki hadni tanlashning o'zidan $g_i = P_i < P_j$). Keyin, o'zining berilgan birinchi i hadlarida P — maksimal, Q — minimal o'rin almashtirish bo'ladi, chunki P da qolgan hadlar kamayish, Q da esa o'sish tartibida keladi. Ni-



4-rasm. • — P o'rin almashtirish; o — Q o'rin almashtirish.

hoyat, R o'rin almashtirish P va Q o'rtasida yotgan bo'lsa, uning dastlabki $i-1$ hadlari P va Q ning dastlabki hadlari bilan mos tushadi, r_i had esa p_i ga yoki g_i ga teng bo'ladi, chunki birinchi $i-1$ band hadlarda P_i va g_i o'rtasida joylashgan sonlar bo'lmaydi. Agar $p_i = r_i$ va $P \leq R$ bo'lsa, $P=R$ (chunki P_1, P_2, \dots, P_i berilganlarda P maksimal), agar $r_i = g_i$ bo'lsa, xuddi shunday $R=Q$ bo'ladi.

Dasturda bu quyidagicha amalga oshiriladi. Joriy o'rin al-

mashtirish uchun P massiv hosil qilinadi, u birinchi o'rin almash-tirish $P=(1,2,\dots,m)$ bilan to'ldiriladi va berilgan A massiv had-larining tegishli o'rin almashtirilishi chop etiladi.

Navbatdagi P o'rin almashtirish hosil qilingan va unga mos A o'rin almashtirish chop etilgan bo'lsin. i maksimal raqamga ega bo'lgan $P_i < P_{i+1}$ element izlanadi. Agar u bo'lmasa, P o'rin almash-tirish oxirigisi bo'ladi. Aks holda $P_i < P_j$ shart bajariladigan $j < i$ eng katta raqam izlanadi. $P_i < P_j$ elementlarning o'rni almashtiriladi, shundan keyin $P_{i+1}, P_{i+2}, \dots, P_m$ ketma-ketlikda tartib teskarisiga o'z-garadi: buning uchun P_{i+1} va P_m , keyin esa P_{i+2} va P_{m-1} , va hokazolar o'rinlarini almashtirishadi.

Navbatdagi o'rin almashtirishni hosil qilish shu bilan tugaydi va A massivning tegishli o'rin almashtirilishi chop etiladi.

Dastur

Program O'rin almashtirish;

const mm=100;

var

m, i, j, k, n : integer;

A, P : array [1..mm] of integer;

begin

readln (m);

write ('m=',m);

for i:=1 to m do

begin

readln (a[i]);

p[i]:=i

end;

for i:=1 to m do write (a[i],");

writeln;

for i:=m-1 downto 1 do if (p[i]<p[i+1]) then

begin

n:=p[i];

for j:=m downto i do

if (n<p[j]) then

begin

p[i]:=p[j]; p[j]:=n;

k:=1;

while i+k<m-k+1 do

begin

```

n:=p[i+k];
p[i+k]:=p[m+1-k];
p[m+1-k]:=n;
k:=k+1;
end;
j:= i;
end;
for i:=1 to m do write (a[p[i]],");
writeln;
end;
end.

```



246-masala
 Algoritm. «Tub sonlar»

Topilgan tub sonlar uchun hisoblashni tezlashtirish maqsadida jadval hosil qilish foydali. Navbatdagi nomzodni faqat shu jadvaldagi sonlarga bo‘linishini tekshirish kerak. Juft sonlar, tabiiy, qaralmaydi. Jadval kamida $\sqrt{M/2}$ sonlar uchun ishlatilishi mumkin. Ya’ni 1000 ta sonli jadval 4000000 gacha bo‘lgan sonlarni chop etish uchun yetarli.

Algoritm va dasturni, 3 ga bo‘linadigan sonlarni e’tibordan chetda qoldirib yaxshilash mumkin.

Dastur

```

Program Tub_son;
const n=200;
label 1,2;
var
  m, i, j, k, q : integer;
  P : array [1..n] of integer;
begin
  write ('M=');
  readln (m);
  write ('N=',N);
  if (m >= 2) then writeln (2);
  if (m >= 3) then writeln (3);
  k:=1; P[k]:=3; i:=5;
  while i <= m do
    begin
      for j:=1 to k do

```



```

begin
    g:=p[j];
    if (g*g>i) then goto 1;
    if (i mod g=0) then goto 2;
end;
if k=n then I=m-1
else
1: begin
    writeln (i);
    if k<=n-1 then
        begin
            k:=k+1;
            p[k]:=I;
        end;
end;
2: i:=i+2;
end;
end.

```



247-masala
 Algoritm. «Teng sonlarni izlash»

Bu masalani qoniqarli hal qilish uchun, $i=p$ va $j=g$ hollarda, chalkashmaslik maqsadida, ikki marta taqqoslash uchun bir xil a $[i,j]$ va a $[p,g]$ elementlar juftini olmaslik kerak.

Dastur

```

Program Teng_sonlar;
label 5;
var
    i,j,p,b,g : integer;
    ff       : boolean;
    A : array [1..2,1..15] of real;
begin
    for j:=1 to 15 do
        begin
            write ('A[1, 'j, ']:=');
            readln (a[1,j]);
            writeln ('A[2, 'j, ']:=');
            readln (a[2,j]);
        end;
    end;

```

```

ff:=false;
for i:=1 to 2 do
  for j:=1 to 15 do
    for p:=1 to 2 do
      begin
        if i<p then b:=1 else b:=j+1;
        for g:=b to 15 do
          if (A[i,j]=A[p,g]) then
            begin
              ff:=true;
              goto 5;
            end;
          5: if ff then writeln ('A['i, ', ', 'j, ']=A['p, ', ', 'g, ']);
              else writeln ('___');
        end
      end
end.

```



248-masala

Algoritm. «Raqamlarning berilgan yig'indisi»

Izlanayotgan sonning raqamlariga ko'ra uchlamchi siklini yozish mumkin:

i bo'yicha 0 dan 9 gacha, j bo'yicha 0 dan 9 gacha, k bo'yicha 1 dan 9 gacha.

Unda $i+j+k$ yig'indisini hisoblash va u berilgan songa teng bo'lsa, uch xonali: $M=i+10j+100k$ sonni chop etish kerak. Lekin bu yaxshi yechim emas. Unda mumkin bo'lgan 100 ta sikl o'rniga 900 ta sikl bajariladi.

Qabul qilinishi kerak bo'lgan yechim i va j bo'yicha ikkilamchi siklga ega, K esa « n » ning berilgan yig'indisi bilan hisoblanadi.

$$K = n - i - j$$

$1 \leq k \leq 9$ tekshirishni kiritib, taklif etilgan dasturga kelish mumkin. Dasturni $n \leq 27$ shartni tekshirish bilan boshlab, $n < 18$ holda hisoblashning boshqa tezlanishlarini kiritib, dasturni takomillashtirish mumkin.

Dastur

```
Program Raqam_beril_summasi;
```

```
var
```

```
  i,j,k,n : integer;
```

```
begin
```

```

writeln ('n:= ',n);
readln (n);
if n in [1..27] then
for i:=0 to 9 do
    begin
        for j:=0 to 9 do k:=n-i-j;
            if k in [1..9] then writeln
                (i+10*j+100*k);
        end
    end.
end.

```



249-masala
 Algoritm. «Funksiya»

$f(n)$ formulani topish qiyinligidan va n argumentining har bir kamayish qadamida, ko'zdan kechirilish kerak bo'lgan, funksiyalar soni o'sayotganday bo'lib ko'rinishidan, masalani yechish mumkin emasday bo'lib tuyuladi. Haqiqatan ham, $f(2n+1)$ funksiyadan chiqilsa, ikkita funksiya hosil qilinadi:

$$f(2n+1) = f(n) + f(n+1)$$

Ikkita n va $n+1$ argumentlardan bittasi toq. Keyingi qadamda u yana ikkita funksiyani tug'diradi va hokazo. Lekin ikkinchi qadamni bajara turib, bu funksiya avvalgiday uchta emas, balki ikkita va hamma vaqt ikkita bo'lib qolayotganini sezish mumkin.

Dasturni hosil qiluvchi bu holni isbotlash uchun, berilayotgan bir argumentli $f(n)$ funksiya bilan bir qatorda quyidagi n, i, j — uch funksiya kiritiladi:

$$g(n, i, j) = if(n) + if(n+1)$$

Uning uchun rekurrent formulalar: $g(2n, i, j) = g(n, i+j, j)$, $g(2n+1, i, j) = g(n, i, i+j)$ oson tekshiriladi. Endi izlanayotgan $f(n)$ qiymatni quyidagicha yozish mumkin:

$$f(n) = g(n, 1, 0),$$

rekurrent formulalarini ko'p marta qo'llab, g funksiyaning birinchi argumentini nol qilish va $f(n) = g(n, 1, 0) = \dots = f(n) = g(0, i, j) = j$ ni hosil qilish mumkin. Dasturni yozishda til talab qilayotgan rasmiylashtirish ishlarini bajarish qoldi, xolos.

Dastur

```
Program Funksiya;  
var  
    i, j, n : integer;  
begin  
    readln (n);  
    writeln ('n:=',n);  
    i:=1; j:=0;  
    while n > 0 do  
        begin  
            if mod 2=0 then i:=i+j else j:=j+i;  
            n:=n div 2;  
        end;  
    writeln(j);  
end.
```



250-masala

Algoritm. «To'rtburchaklar».

Bu masala «bir fikrlash»da yechiladi: to'rtburchaklarning shimoliy-g'arbiy burchaklari (boshqacha aytganda — yuqori chap) qancha bo'lsa, to'rtburchaklar shuncha. Faqat burchak chegarada bo'lgan holda chalkashilmasa bo'lgani. Bu qiyinchilik dasturlarda har xil hal qilinishi mumkin. Faqat dasturda $i > 1$ and $a[i-1, j] = 0$ turdagi ifodani ishlatmaslik kerakligini ogohlantiramiz. Chunki, agar indeksatsiya birdan boshlansa, $i = 1$ da bunday ifoda sintaktik xato bo'lib hisoblanadi.

Dastur

```
Program To'rtburchak;  
const nn=100; mm=100;  
var  
    i, n, s, j : integer;  
    a : array [0..mm, 0..nn] of byte;  
begin  
    writeln ('m,n=');  
    readln (m,n);  
    for i:=0 to m do  
        for j:=0 to n do  
            if (i=0) or (j=0) then a[i,j]:=0 else  
                begin
```

```

        writeln ('a[';i;',';j;']:=');
        readln (a[i,j]);
    end;
s:=0;
for i:=1 to m do
    for j:=1 to n do
        if (a[i,j]=1) and (a[i-1,j]+a[i,j-1]=0) then s:=s+1;
    writeln(s);
end.

```



251-masala
Algoritm. «Bit-revers»

Aniqlik uchun m va $B(m)$ ning bir nechta qiymatlarini o'nlik va ikkilik sanoq tizimlarida yozamiz:

Tizim	10	2	2	10
m va $B(m)$ qiymatlari	512	1000000000	0000000001	1
	513	1000000001	1000000001	513
	514	1000000010	0100000001	257
	515	1000000011	1100000001	769
	516	1000000100	0010000001	129

	1023	1111111111	1111111111	1023

Jadvaldan ko'rinib turibdiki, « m » sonining bosh ikkilik razryadi 512 ni ifodalamoqda va unga, 1 miqdorini ifodalovchi, $B(m)$ sonining kichik razryadi mos kelmoqda. Keyingi razryadlar uchun bu 256 va 2.. va hokazo miqdorlar bo'ladi. Hamma vaqt « m » sonida, $m \geq 512$ bo'lgani uchun bosh razryadning bir soni bor. Uni, « m »ni $m-512$ ga almashtirib, yo'q qilamiz va birni $B(m)$ soniga kiritamiz.

Keyingi razryadda, agar yangi qiymat $m \geq 256$ bo'lsa, « m »sonida bir bo'ladi. Agar u bo'lsa, uni yo'q qilamiz va $B(m)$ ga 2 sonini kiritamiz. Navbatdagi birming bor-yo'qligini tekshiramiz va hokazo. Ish « m » soni nolga aylanganida to'xtatiladi. Bu ishlarni o'z ichiga olgan dastur o'zining hisoblashlari uchun $512 \cdot 10$ siklni sarflaydi.

$B(m)$ sonlari bo'yicha $B(m+1)$ sonini yasovchi dastur tezroq ishlaydi. $B(m)$ qiymatlarining ikkilik yozuvlarini ko'zdan kechirib, quyidagini tushunib olish mumkin: $B(m+1)$ ni yasashda $B(m)$ sonini ikkilik yozuvida chapdan o'ngga tomon, birlarni nollar bilan birinchi

nolgacha almashtira borib (ya'ni, $B(m)$ dan 512,256... larni ayira borib), siljish kerak. Birinchi nolni bir bilan almashtirish kerak, $B(m+1)$ soni hosil bo'ladi. Bu dastur, sonning yarmini hisoblashga bittadan sikl, choragini hisoblashga ikkitadan sikl, sakkizdan birini hisoblashga uchtadan sikl va hokazo, sarflaydi. Hammasi bo'lib $512 \cdot 2$ tadan kam sikl ketadi. Keltirilgan dastur shunga asosan tuzilgan.

Dastur

```

Program Bit_revers;
var
    m, b, k : integer;
begin
    k:=512; m:=1;
    writeln (m);
    while m < 1024 do
        begin
            while m >= k do
                begin
                    m:=m-k; k:=k div 2;
                end;
            m:=m+k; k:=512;
            writeln (m);
        end
    end.

```



252-masala Algoritm. «Inversiya»

P massivni nollar bilan tozalaymiz. Navbatdagi $i = 1, 2, \dots, n$ va T sonni olamiz. $T+1$ da nol elementlar uchraguncha P_1, P_2, \dots elementlar bo'ylab o'tamiz va ularning oxirgisiga i sonini yozamiz.

Dastur

```

Program Inversiya;
const nn=100;
var
    i, n, j, k : integer;
    P, T : array [1..nn] of integer;
begin
    writeln ('n=', n);

```

```

readln (n);
for i:=1 to n do
  begin
    writeln ('T[',i,']=');
    readln (T[i]);
    P[i]:=0;
  end;
for i:=1 to n do
  begin
    j:=0; k:=0;
    repeat
      k:=k+1;
      if P[k]:=0 then j:=j+1;
    until j>T[i];
  end;
  P[k]:=i;
for i:=1 to n do
  writeln (P[i]);
end.

```



253-masala

Algoritm. «Qo'shiluvchilarga ajratish»

Qo'shiluvchilar o'rnini almashtirishda ajratishlarning takrorlanmasligiga erishish uchun «n» sonining shunday natural qo'shiluvchilarga ajralishini ko'rib o'tamizki, $n=m_1+m_2+\dots$, bunda, $m_1 \geq m_2 \geq \dots$

Ajralishni tartibga solamiz:

$$n=m'_1+m'_2+\dots \quad (m')$$

$$n=m''_1+m''_2+\dots \quad (m'')$$

— ikkita ajratish. Agar birinchi tengmas $m'_i \neq m''_i$ juftdayoq, $m'_i > m''_i$, ya'ni hamma $j < i$ va $m'_j > m''_j$ da $m'_j \neq m''_j$ bo'lsa, (m') ajratish (m'') ajratishdan oldin keladi, deb hisoblaymiz. Shu qoidaga ko'ra, $n=5$ sonini qo'shiluvchilarga ajratish quyidagi tartibda bo'ladi:

$$5=5,$$

$$5=4+1,$$

$$5=3+2,$$

$$5=3+1+1,$$

$$5=2+2+1,$$

$$5=2+1+1+1,$$

$$5=1+1+1+1+1.$$

Berilgan N sonining qo‘shiluvchilarga ajratishning hamma ko‘rinishlarini hosil qilish uchun ishni $n=n$ ajratishdan boshlaymiz. Agar navbatdagi ajratish

$$n = m_1 + m_2 + \dots + m_i \quad (*)$$

hosil qilingan bo‘lsa, uni chop etamiz va bevosita undan keyin keluvchisini tuzishga o‘tamiz. Buning uchun massiv bo‘yicha o‘ngdan chapga tomon siljib borib, m_1, m_2, \dots, m_k ni, « m_k » ning birdan katta qo‘shiluvchisigacha ko‘rib boramiz. Agar bunaqasi uchramasa, hosil bo‘lgan (*) ajratish oxirgisi va ish tugagan, $m_k > 1$ qo‘shiluvchi topilgan bo‘ladi. m_k ni birga kamaytirib va undan keyin keluvchi birlik qo‘shiluvchilarni olib tashlab, butun yig‘indini $S = 1 + i - k$ miqdorda kamaytiramiz. Endi $j = k + 1, k + 2, \dots$, uchun m_j ning yangi ketma-ket qo‘shiluvchilarini aniqlaymiz. Agar $S > m_k$ bo‘lsa, $m_j = m_k$, deb olamiz va S ni m_k miqdorga kamaytiramiz. Agar $S \leq m_k$ bo‘lsa, $m_j = S$ deb olamiz va shu bilan n ning qo‘shiluvchilarga ajratishning yangi ko‘rinishini tugatamiz.

Dastur

Program Qo‘shiluvchilarga_ajratish;

const NN=100;

label ra,br;

var

n, i, k, t, s : integer;

M : array [1..NN] of integer;

begin

writeln ('n:=,');

readln (n);

writeln (n);

m [1]:=n; k:=1; i:=1;

ra: t:=m[k]-1 ; s:=t+i-k+1;

for i:=k to n do

if s>t then

begin

m[i]:=t; s:=s-t

end

else

begin

m[i]:=s;

goto br

end;


```

br: for k:=1 to 1 do writeln (m[k], ' ');
    writeln;
    for k:=i downto 1 do
        if m[k]>1 then goto ra
end.

```



254-masala Algoritm. «O'ng katta»

Yechish uchun ikki xil fikr ishlatiladi. Birinchidan, $A[1:n]$ massiv oxiridan ko'zdan kechiriladi; ikkinchidan, A massiv keyingi ko'zdan kechirilishida, uning elementlarini almashtirish uchun kerak bo'ladigan elementlaridan $B[1:n]$ massiv hosil qilinadi. Aniqroq qilib aytganda, $A[i]$ massivning navbatdagi elementi ko'zdan kechirilayotganda, B massivda hamma $a[i+1, j-1]$ elementlardan katta $a[j](j>i)$ massiv elementlari yotgan bo'ladi. Elementlar A massivda qanday tartibda joylashgan bo'lsa, B massivda ham xuddi shunday tartibda joylashgan bo'ladi, faqat $B[k:n]$ kesmani to'ldira borib o'ng tomonga siljigan bo'ladi.

A va B massivlar bir vaqtda qayta ishlanadi. Avval $k=n: b[k]=a[n]:a[n]=0$, deb olinadi. Navbatdagi $a[i]$ element qaralayotganda, $A[i+1:n]$ sohadagi hamma elementlar almashtirilgan va $B[k:n]$ massiv kerakli tarzda to'ldirilgan bo'ladi. $B[k:n]$ massivda birinchi $b[j]>a[i]$ element izlanadi. Agar shunday j raqam topilsa, $k=j-1; b[k]=a[i]:a[i]=b[j]$, bajariladi. Agar, shunday j raqam bo'lmasa, $k=n:b[k]=a[i]:a[i]=0$, deb olinadi. Ikkala holda yangi $B[k:n]$ kesma e'lon qilingan o'z xossalarini saqlaydi. Har bir $a[i]$ element $b[j]$ element bilan taqqoslanayotganda yoki $a[i]$ element B ga kiritiladi yoki $b[j]$ element B dan yo'q qilinadi. Shunday tarzda A massivning har bir elementi birinchi turdagi (u B ga kiritiladigan holda) bitta taqqoslashni va bittadan ko'p bo'lmagan — ikkinchi turdagi (B dan yo'qotilgan holda) taqqoslashni tug'diradi. Demak, taqqoslashlar $2n$ dan, amallar esa n tartibdan ko'p emas.

Dastur

```

Program O'ng_katta;
const NN=100;
label 1,2;
var
    j, i, k, n : integer;

```

```

ff : boolean;
A,B : array [1..NN] of real;
begin
  writeln ('N:=');
  readln (n);
  for i:=1 to n do
    begin
      write ('A[i,]:=');
      readln (A[i]);
    end;
    B[n]:=A[n]; A[n]:=0; k:=n;
  for i:=n-1 downto 1 do
    begin
      j:=k; ff:=false;
      while (j<=n) and not ff do
        begin
          ff:=A[i]<B[j];
          j:=j+1;
        end;
      if j < n then
        k:=j-1-1;
        B[k]:=A[i];
        A[i]:=B[j-1];
      end
      else
        begin
          K:=n; B[k]:=A[i]; A[i]:=0;
        end;
    end;
  for i:=1 to n do writeln (A[i]);
end.

```



255-masala
 Algoritm. «Rukzak»

30 kg.dan og'ir bo'lgan predmetlar olib tashlanib, qolganlari ma'lum bir tartibda joylashtirilgach, variantlar shajarasini quyidagicha aniqlaymiz. Navbatdagi $I=1,2,\dots,n$ yo'lda i raqamli predmetni qaraymiz, i yo'ning j variantlari esa hamma vaqt ikkita bo'ladi: $j=0$ predmetni olish, $j=1$ predmetni olmaslikni bildiradi. Tarmoqlari n uzunlikka teng ikkilamchi daraxt hosil bo'ladi.

Berilgan $A[1:n]$ va $B[1:n]$ massivlardan tashqari $P[1:n]$ massiv va bir nechta o'zgaruvchi kiritamiz:

i – navbatdagi predmet raqami;
 T – rukzakdagi predmetlar og'irligi;
 z – rukzakdagi predmetlarning jami qiymati;
 ZM – ko'rilgan variantlarning maksimal qiymati;
 $k \leq i$ predmet rukzak olinsa, $P[k]=0$;
 $k \leq i$ predmet rukzakka olinmasa, $P[k]=1$.

Boshda i, S, Z, ZM nolga tenglashtirib olinadi. Variantlarni ko'zdan kechirishda predmetning (va uning hamma davomi) qiziqish tug'dirmasligi aniq bo'lishi bilan ko'rib chiqishni to'xtatish muhimdir. Oldinga harakat qilishda (agar $S+A[i]<30$ bo'lsa), predmetni rukzakka qo'yishga intilamiz. Bu holda biz chap tarmoq bo'yicha boramiz:

$$S=S+A[i] : Z=Z+B[i] : P[i]=0.$$

Agar predmetni qo'shish mumkin bo'lmasa, uni olamiz (ya'ni chappa ketuvchi variant tarmoqlarini tashlab borib, o'ng tarmoq bo'yicha harakatlanamiz) va $P[i]=1$ deb olamiz. Ikkala holda ham eng oxirgi predmet ko'rilmaguncha oldinga harakatni davom ettiramiz.

Agar hamma variantlar ko'zdan kechirilgan bo'lsa, variant hosil qilindi. U ZM bilan taqqoslanadi:

$$\text{if } ZM < Z \text{ then } ZM = Z$$

va oxiriga qarab harakat boshlanadi.

Oldinga harakat qilishda, olingan ketma-ket keluvchi predmetlarning hamma guruhi o'tkazib yuboriladi (ularda $P[i]=0$), chunki bitta shu guruhdagi o'zgarish rukzakdagi predmetlarning jami qiymatini tushiradi, xolos. Ko'rilgan predmetlar rukzakdan yo'l-yo'lakay olib tashlanadi:

$$\text{if } P[i]=0 \text{ then } S=S-A[i] : Z=Z-B[i].$$

Shundan keyin, oldin olinmagan predmetlarning butun guruhi o'tkazib yuboriladi (ularda $P[i]=1$), chunki bu guruhdagi o'zgarish oldin baholanishi kerak bo'lgan chap tarmoqqa olib keladi.

Qisqa qilib aytganda, biz $P[i]=0$ va $P[i+1]=1$ ni hosil qiladigan shunday raqamga erishgunimizcha, oxiriga qarab harakat qilamiz. Bunday harakatda rukzakdan unda mavjud bo'lgan predmetlar olib tashlanadi. Agar kerakli i bo'lmasa, ish tugatiladi.

Dastur

```

Program Rukzak;
const NN=100;
      T=30;
label 1;
  
```

var

i, s, z, zm, n : integer;
A, B : array [1..NN] of real;
P : array [1..NN] of boolean;

begin

writeln ('N:=');

readln (n);

for i:=1 to n do

begin

writeln ('A[',i,']:=');

readln (A[i]);

writeln ('B[',i,']:=');

readln (B[i]);

end;

S:=0; Z:=0; ZM:=0; i:=0;

1: for i:=i+1 to n do

it S+A[i]>=T then P[i]:=false else

begin

S:=S+A[i];

Z:=Z+B[i];

P[i]:=TRUE

end;

if zm < z then zm:=z;

for i:=n-1 downto 1 do

begin

if P[i+1] then

begin

S:=S-A[i+1];

Z:=Z-B[i+1];

end;

if P[i] and not P[i+1] then

begin

S:=S-A[i];

Z:=Z-B[i];

P[i]:=False;

Goto 1

end;

end;

writeln (ZM)

end.

FOYDALANILGAN ADABIYOTLAR

1. М. Зелкович, А. Шоу, Дж. Гэннон. Принципы разработки программного обеспечения. Пер. с англ. М., «Мир», 1982.
2. Дж. Фокс. Программное обеспечение и его разработка. Пер. с англ. М., «Мир», 1985.
3. Практическое руководство по программированию. Пер. с англ. М., «Радио и связь», 1986.
4. Н. Вирт. Алгоритмы и структуры данных. Пер. с англ. М., «Мир», 1989.
5. А.Л. Брудно, Л.И. Каплан. Московские олимпиады по программированию. 2-е издание. М., «Наука», 1990.
6. В.В. Липаев. Проектирование программных средств. М., 1990.
7. В.Ф. Очков, Ю.Ю. Пухначев. 128 советов начинающему программисту. М., 1991.
8. А.В. Файсман. Профессиональное программирование на Турбо Паскале. Т., Информ Экс — Корпорейшн, 1992.
9. Н.Б. Культин. Программирование в *Turbo Pascal 7.0* и *Delphi* / Второе издание, переработанное и дополненное. СПб, БХВ — Санкт-Петербург, 1999.
10. Н.Б. Культин. Программирование на *Object Pascal* в *Delphi 5*. СПб, БХВ — Санкт-Петербург, 1999.
11. А.Н. Марченко. Программирование в среде *Turbo Pascal 7.0*. К., Век+, М., ДЕСС, 1999.
12. В.В. Фаронов. *Turbo Pascal 7.0*. М., Нолидж, 2000.
13. Н.Б. Культин. *Turbo Pascal* в задачах и примерах. СПб, БХВ — Санкт-Петербург, 2001.

MUNDARIJA

Kirish	3
--------------	---

I qism. NAZARIY MA'LYMOTLAR

1.1. Turbo Pascalda dastur tuzish uchun kerakli ma'lumotlar	4
1.1.1. Dastur tasnifi	4
1.1.2. Ma'lumotlarning asosiy turlari	4
1.1.3. Massivlar	5
1.1.4. Yozuvlar	5
1.1.5. Ko'rsatmalar (operatorlar)	6
1.1.6. Funksiyalarni e'lon qilish	9
1.1.7. Protseduralarni e'lon qilish	9
1.2. Turbo Pascal rezerv so'zlari, direktivalari, standart funksiya va protseduralari	9
1.2.1. Rezerv so'zlar va direktivalar	9
1.2.2. Turbo Pascal direktivalari	10
1.2.3. Funksiya va protseduralar	10
1.2.3.1. Matematik funksiyalar	11
1.2.3.2. O'zgartirishning funksiya va protseduralari	12
1.2.3.3. Satr va belgilar bilan ishlash uchun funksiyalar va protseduralar	13
1.2.3.4. Fayllar bilan ishlash uchun protsedura va funksiyalar	14
1.2.3.5. Boshqa funksiyalar	15
1.3. O'zgaruvchilarning xotirada saqlanishi	19
1.4. EHMda Paskal tilida yozilgan dasturlar bilan ishlash	20

II qism. MASALALAR SHARTLARI

2.1. Quyi murakkablikdagi masalalar	25
2.1.1. O'zgaruvchilarni e'lon qilish	25
2.1.2. O'zlashtirish ko'rsatmasi	26
2.1.3. Chiqarish ko'rsatmasi	26
2.1.4. Kiritish ko'rsatmasi	28
2.1.5. Chiziqli tasnifli dasturlar	29
2.1.6. Tanlov (If) ko'rsatmasi	32
2.1.7. CASE ko'rsatmasi	36
2.1.8. FOR ko'rsatmasi	37
2.1.9. REPEAT ko'rsatmasi	43
2.1.10. WHILE ko'rsatmasi	44
2.1.11. Belgilar va qatorlar	45

2.1.12. Massivlar	47
2.1.13. Funksiyalar	52
2.1.14. Protseduralar	55
2.1.15. Fayllar	56
2.1.16. Rekursiya	60
2.2. Yuqori murakkablikdagi masalalar	61

III qism. MASALALAR DASTURLARI

3.1. Quyi murakkablikdagi masalalar dasturlari	69
3.1.1. Chiqarish ko'rsatmali dasturlar	69
3.1.2. Chiziqli tasnifli dasturlar	70
3.1.3. <i>IF</i> tanlov ko'rsatmali dasturlar	73
3.1.4. <i>CASE</i> ko'rsatmali dasturlar	81
3.1.5. <i>FOR</i> ko'rsatmali dasturlar	84
3.1.6. <i>REPEAT</i> ko'rsatmali dasturlar	96
3.1.7. <i>WHILE</i> ko'rsatmali dasturlar	98
3.1.8. Belgilar va qatorlar ishlatilgan dasturlar	99
3.1.9. Massivlarga oid dasturlar	108
3.1.10. Funksiyalar ishtirokidagi dasturlar	128
3.1.11. Protsedurali dasturlar	134
3.1.12. Fayllarga oid dasturlar	135
3.1.13. Rekursiyali dasturlar	145
3.2. Yuqori murakkablikdagi masalalar algoritmlari va dasturlari	149
Foydalanilgan adabiyotlar	213

Yu57 Yunusova M. J., Rahimov A.B. Dasturlash bo'yicha praktikum. Kasb-hunar kollejlari uchun o'quv qo'llanma. T.: «ILM ZIYO», 2006. — 216b.

BBK 32.973-018ya722

MUSLIMA JAMOLOVNA YUNUSOVA,
AZIM BAGIROVICH RAHIMOV

DASTURLASH BO'YICHA PRAKTIKUM

Kasb-hunar kollejlari uchun o'quv qo'llanma

Toshkent — «ILM ZIYO» — 2006

Muharrir *I. Usmonov*
Rassom *Sh. Qahharov*
Texnik muharrir *F. Samadov*
Musahhih *F. Temirxo'jayeva*

2006-yil 14-martda chop etishga ruxsat berildi. Bichimi 60x90¹/₁₆,
«Tayms» harfida terilib, ofset usulida chop etildi. Bosma tabog'i 13,5.

Nashr tabog'i 14,0 b. t. 3500 nusxa. Buyurtma № 51

Bahosi shartnoma asosida.

«ILM ZIYO» nashriyot uyi. 700129, Toshkent, Navoiy ko'chasi, 30-uy.
Shartnoma № 52—2005.

O'zbekiston Matbuot va axborot agentligining G'afur G'ulom nomidagi
nashriyot-matbaa ijodiy uyida chop etildi.

700128, Toshkent, U.Yusupov ko'chasi, 86-uy.