

**TÜRKMENISTANYŇ BILIM MINISTRLOGI
TÜRKMEN POLITEHNIKI INSTITUTY**

B.Atajanow, O.Nurgeldiýew, G.Çaryýew, Ö.Çowdurow

PROGRAMMIRLEMEGIŇ TILSIMATY

Aşgabat 2010

SÖZBAŞY

Garaşsyz, baky Bitarap Türkmenistan döwletimizde geljegimiz bolan ýaşlaryň dünýäniň iň ösen talaplaryna laýyk gelýän derejede bilim almagy üçin ähli işler edilýär.

Hormatly Prezidentimiz döwlet başyna geçen ilkinji gününden bilime, ylma giň ýol açdy, Türkmenistan ýurdumyzda milli bilim ulgamyny kämilleşdirmek boýunça düýpli özgertmeler geçirmäge girişdi.

Türkmenistanyň Prezidenti Gurbanguly Berdimuhamedowyň «Türkmenistanda bilim ulgamyny kämilleşdirmek hakynda» 2007-nji ýylyň 15-nji fewralyndaky Permany bilim ulgamyndaky düýpli özgertmeleriň başyny başlady.

Häzirki wagtda milli bilim ulgamyndaky döwrebap özgertmeler ýaş nesliň ýokary derejede bilim almagyna we terbiýelenmegine, giň dünýägaraýyşly, edep-terbiýeli, tämiz ahlakly, kämil hünärmenler bolup ýetişmeklerine uly ýardam edýär.

Türkmenistanyň Prezidenti Hormatly Gurbanguly Berdimuhamedowyň baştutanlygynda Täze Galkynyş we Beýik özgertmeler zamanasynda halk hojalygynyň ähli pudaklarynda dünýäde iň kämil tehnologiýalar ornaşdyrylýar. Täze tehnologiýalar bolsa ýurdumyzdaky ähli edaradyr kärhanalarda kompýuterleriň giň gerim bilen ulanylmagyny talap edýär. Geljekki hünärmenler kompýuterler arkaly edaralary dolandyrmagyň usullaryny we ýollaryny bilmelidirler. Gelejekki hünärmen kompýuterleşdirilen edaralarda işiň gurnalys ýagdaýyna obýektiw baha bermegi, soňra onuň netijeleri boýunça esaslandyrylan dolandyrys çözgütlerini kabul etmegi, geljege gönükdirilen, netijeli çözgütleri kabul etmek bilen baglanyşykly näbelliligi peseltmegi, kompýuterleri ulanmaklygyň maksada laýyklygyny, Windows we Linux ýaly operasion sistemalaryny doly öwrenmekligi, kompýuterlerde programma üpjünçiligi

dolandyrmaklygy, kompýuter ulgamyndaky näsazlylyklary düzetmegi, maglumatlary goramaklygy başarmalydyrlar.

Okuw kitaby Täze Galkynyş we Beýik özgertmeler zamanasynda ýokary bilimli hünärmenleri taýýarlamaklyga bildirilýän talaplary göz önünde tutup taýýarlanyldy.

Ders talyplaryň programmalaşdyrmagyň haýsy hem bolsa bir dilini özleşdirenlerinden soň geçilýär. Adatça programmalaşdyrma dilleriniň biri Kompýuter we programmirmek dersinde geçilýär. Dersde dürli meseleleri çözmekde ulanylýan algoritmler özleşdirilýär. Hususanda talyplaryň programmalaşdyrmakda wajyp orun tutýan komekçi programmalary ulanmak endiklerini ösdürmek, massiwleri tertipleşdirmekdäki ulanylýan algoritmleri, dürli matematiki meseleleri çözmegiň algoritmlerini, senenamalar bilen baglanyşykly meseleleri çözmegiň usullaryny öwrenmekleri üçin gerekli maglumatlar berilýär. Şeýle hem kitapda simwollar we setir ululyklar bilen işlemekde ulanylýan esasy usullar, maglumatlaryny gaýtadan işlemeklige degişli durmuşda duş gelýän käbir meseleler, oýunlary programmalaşdyrmak meseleleri, grafiki ekrany dolandyrmagyň usullary, kompýuter portlaryny ulanmagyň usullary beýan edilýär.

PROGRAMMALAŞDYRMAKDA KÖMEKÇİ PROGRAMMALARYŇ ULANYLYŞY

§1. Kömekçi programmalar. Kömekçi programmalaryň görnüşleri.

Käbir meseleler çözülende parametrleriň dürli bahalarynda şol bir algoritmi birnäçe gezek gaýtalap ýerine ýetirmeli bolýar. Beýle ýagdaýda programmanyň şol algoritme degişli bölegini aýratyn bölüp alyp, oňa gerek wagty programmanyň islendik böleginden ýüzlenmek mümkindir.

Programmanyň özbaşdak programma birligi hökmünde ulanyp bolýan bölegine kömekçi programma diýilýär. TURBO-PASCAL algoritmik dilinde kömekçi programmanyň iki görnüşi ulanylýar:

1. Funksiýalar;
- 2). Proseduralar.

Eger hasaplama ýerine ýetirilenden soň alnan netije diňe bir sany baha bolsa, onda onuň ýaly hasaplamaný funksiýa görnüşinde ýazyp bolýär. Funksiýa hem üç bölekden durýar:

1. Sözbäşy
2. Yglan ediş bölümi
3. Funksiýanyň göwresi

Funksiýa umumy görnüşde aşakdaky ýaly ýazylýar:

FUNCTION F(q1:T1;q2:T2;...,qn:Tn):T; {Funksiýanyň sözbäşy}

<Yglan ediş bölümi>

BEGIN

P1;

P2;

...{Funksiýanyň göwresi}

F:=...

END;

Bu ýerde F-funksiýanyň ady; $q_i (i = 1, n)$ - parametrleriň sanawy; $T_i (i = 1, n)$ -degişlilikde olaryň tipleri; T- ol F funksiyanyň adynyň tipi ýa-da başgaça netijäniň tipi; P1, P2,...-funksiýanyň göwresini emele getirýän operatorlar. Yglan ediş bölümünde diňe funksiyanyň çäginde ulanylýan parametrlrler we olaryň tipleri görkezilýär. Funksiya özünde hemişelikler bölümüni, belgiler bölümüni, tipler bölümüni, üýtgeýän ululyklar bölümüni, kömekçi funksiýalary we proseduralary saklap biler. Eger parametrleriň birnäçesi şol bir tipe degişli bolsa, onda olary toparlar, tipi diňe bir gezek ýazmaklyga rugsat edilýär. Meselem:

FUNCTION FAKT(K1,K2,K3 : INTEGER) : REAL;

Eger-de algoritm ýerine ýetirilenden soň alynýan netije birden köp – birnäçe baha bolsa, onda kömekçi programmanyň prosedura görnüşinden peýdalanylýar. Proseduralar umumy görnüşde aşakdaky ýaly ýazylýar:

```
PROCEDURA F(parametrleriň sanawy);
{Proseduranyň sözbaşy}
<Yglan ediş bölümü>
{bu bölümde lokal parametrlrler yglan edilýär}
```

BEGIN

```
P1;
P2;
... {Operatorlar bölümü}
Pn;
```

END;

Bu ýerde F-proseduranyň ady, funksiyanyň adyndan tapawutlanyp, ol bu ýerde hiç-hili baha eýe bolmaýar. Şoňa görä-de onuň üçin tip kesgitlenmeýär.

§ 2. Formal parametrler we parametrleriň iş ýüzündäki bahalary. Lokal we global parametrler.

Kömekçi programma kesgitlenende onuň sözbaşysynda parametrleriň sanawy getirilýär. Bu sanawdaky parametrlere kömekçi programmanyň formal parametrleri diýilýär. Formal parametrleriň sanawynda kömekçi programma ýüzlenmek üçin gerek bolan parametrler we olaryň tipleri görkezilýär.

Programmada kömekçi programma ýüzlenilende onuň adyndan soň kömekçi programmada ulanylýan parametrleriň bahalary ýa-da atlary görkezilýär. Olara parametrleriň iş ýüzündäki bahalary diýilýär.

Funksiýa programmanyň islendik ýerinden ýüzlenmek üçin baha bermek operatorynyň sag tarapyndaky aňlatmada onuň adyny we ýaýyň içinde formal parametrleriň iş ýüzündäki bahalaryny görkezmek ýeterlikdir.

$F(b_1, b_2, \dots, b_n)$;

bu ýerde F -funksiýanyň ady; b_1, b_2, \dots, b_n – deňişlilikde q_1, q_2, \dots, q_n – formal parametrleriniň iş ýüzündäki bahalary. Formal parametrleri bilen onuň iş ýüzündäki bahalarynyň sany deň bolmaly we olar deňişlilikde tipleri boýunça gabat gelmeli.

Mysal:

$$C_n^m = \frac{n!}{m!(n-m)} \{n, m \in \mathbb{N}, n > m\} \text{ formula boýunça}$$

utgaşmalaryň sanyny hasaplamak üçin programma düzmelili.

Faktorial hasaplamagy bölek programma – funksiýa

görnüşinde ýazmaly.

PROGRAM UTGSANY;

USES CRT;

VAR CNM : REAL;

N, M, L : INTEGER;

FUNCTION FACT(K : INTEGER;) :

INTEGER;

VAR P, I : INTEGER;

BEGIN

```
P:=1;  
FOR I:=1 TO K DO P:=P*I;  
FACT:=P  
END;
```

BEGIN

```
CNM:=FACT(N)/(FACT(M)*(FACT(N-M)));  
WRITE('Utgaşmanyň sany=',CNM:8:2);  
END.
```

Prosedura aşakdaky ýaly ýüzlenilýär.

$F(b_1, b_2, \dots, b_n);$

Bu ýerde b_1, b_2, \dots, b_n –formal parametrleriň iş ýüzündäki bahalary; edil funksiýadaky ýaly, bu ýerde hem formal parametrler bilen onuň iş ýüzündäki bahalarynyň sany gabat gelmelidir we olar deňişlilikde şol bir tipe deňişli bolmalydyr.

Prosedura funksiýadan tapawutlanyp, birden köp-birnäçe netijäni dolandyryp, kä ýagdaýlarda bolsa, hiç-hili netijäni dolandyрман hem biler. Meselem:

```
PROGRAM A6;  
ESES CRT;  
VAR I: BYTE;  
PROCEDURE LINIA (N: INTEGER);  
VAR  
    I: INTEGER;  
BEGIN  
    CLRSCR;  
    FOR J=1 TO N DO  
        WRITE( '- ' ) END;  
BEGIN  
    FOR I:=1 TO 6 DO  
        BEGIN LINIA (I); WRITELN; END  
END.
```

Görnüşü ýaly, bu ýerde her gezek bölek programma ýüzlenilende yzyna netijä dolanylmaýarda, diňe käbir operatorlaryň toplumy ýerine ýetirilýär. Programmada WRITELN operatory her gezek täze setire geçmek üçin hyzmat edýär.

Funksiýada we prosedurada formal parametrleri islendik tertipde ýerleşdirmek mümkin. Bölek programma ýüzlenilen mahalynda formal parametrler nähili tertipde ýerleşdirilen bolsa, olaryň iç ýüzündäki bahalary hem şol tertipde ýerleşdirilen bolmaly.

Prosedurada formal parametrleriň iki görnüşini bir-birinden tapawutlandyrmak gerek:

- 1) Baha hökmünde ulanylýan parametrler;
 - 2) Üýtgeýän ululyk hökmünde ulanylýan parametrler;
- Baha hökmünde ulanylýan parametrler proseduranyň ýerine

ýetirilmegi üçin zerur bolan başlangyç maglumatlary-

berlenleri kesgitlemek üçin hyzmat edýärler. Olar formal

parametrleriň sanawynda:

$(q_1:t_1;q_2:t_2;...) \text{ ýa-da } (q_1,q_2:t;...)$

görnüýazulýar. Bu ýerde $q_1,q_2,...$ -parametrleriň bahalary, hemişelik, üýtgeýän ýa-da aňlatma görnüşde berlip bilner.

Üýtgeýän ululyk hökmünde ulanylýan parametrler adatça prosedura ýerine ýetirilenden soň, alnan netijelere baha bermek üçin hyzmat edýärler. Proseduradan alynýan maglumatlar şol üýtgeýän ululyklar arkaly esasy programma berilýär. Üýtgeýän ululyk hökmünde ulanylýan parametrler formal parametrleriň sanawynda asakdaky görnüşde ýazylýar:

$(..., \text{VAR } q_1 : T_1, q_2 : T_2, ..., q_n : T_n);$

Bu ýerde q_1, q_2, \dots, q_n - üýtgeýän ululyk hökmünde ulanylýan parametrleriň atlary; T_1, T_2, \dots, T_n - degişlilikde olaryň tipleri.

Programmanyň teksti ýerine ýetirilýän koda ýokardan aşaga yzly yzyna geçirilýär. Şunlukda ütgýän ululyklar, tipler, hemişelikler we bölek programmalar operatorlarda we aňlatmalarda entäk ulanylmanka beýan edilmelidir. Eger şeýle edilmese, onda kompilýator degişli atlary näbelli diýip habar berer. Bu ýagdaýda degişli bölek programmany programmanyň ýazgysynda ýokaryk geçirmeli bolar.

Ýöne kä ýagdaýlarda bir programmada beýleki bir entek yglan edilmedik bölek programmany çagyrmaly bolýar. Onuň üçin şol çagyryljak bölek programmanyň söz başyçynyň yzyndan FORWARD-kömekçi sözünü ýazmaly. Ol bölek programmanyň göwresini bolsa ol ýerde ýazmak hökman dälidir.

Soň ol kömekçi programmanyň göwresini esasy programmanyň islendik ýerinde kesgitlemek bolýar. Öňünden kesgitlenen kömekçi programmanyň kesgitlenýän wagty onuň sözbaşysynda parametrlr görkezilmese hem bolýar. Eger ol parametrlr kömekçi programma kesgitlenýän wagty beýan edilýän bolsa, onda olaryň ýazylyş tertibi we görnüşi edil öňünden kesgitlenen ýerindäki (FORWARD) ýaly bolmalydyr. Bu ýagdaýy aşakdaky mysalyň üsti bilen düşündürmek mümkin:

PROGRAM A5;

...

VAR X,Y: real;

PROCEDURE P1(A:REAL); FORWARD;

PROCEDURE P2(B:REAL);

BEGIN

...

P1(X)

END;

```

PROCEDURE P1;
...
BEGIN
P2(Y)
END;
BEGIN
P2(X);
P1(Y);
...
END.

```

Öňünden beýan etmegi ulanyp aşakdaky ýaly halkalaýyn çagyrmagy hem gurnamak mümkindir.

Procedure a(y:TypeXY); Forward;

Procedure b(x:TypeXY);

Begin

...

a(p); { b prosedura a prosedurany çagyryar }

end;

Procedure a;

Begin

...

b(q); { ýöne a proseduranyň özi hem b-ni çagyryar }

end;

Kömekçi programanyň içinde beýan edilen üýtgeýän ululyklara lokal üýtgeýän ululyklar, esasy programmanyň özünde beýan edilen üýtgeýän ululyklara bolsa global üýtgeýän ululyklar diýilýär. Lokal we global düşüňjeleri üýtgeýän ululygyň hereket edýän ýaýlasyny görkezýärler.

§3. Rekursiýanyň ulanylyşy

TURBO-PASCAL algoritmik dilinde bir kömekçi programmanyň içinde ýene-de şol kömekçi programmanyň özüne ýüzlenmäge rugsat edilýär. Kömekçi programma ýüzlenmekligiň bu görnüşine rekursiw ýüzlenme diýilýär.

Köplenç metematiki formulalary rekursiýa görnüşde ýazyp bolýar. Meselem :

1, eger $n=0$ bolsa

$X^n =$

X^2 , eger $n>0$ bolsa.

Ýa-da

1, eger $n=0$ bolsa,

$N! = N (N^{-1}) !$, eger $n>0$ bolsa we ş. m.

Mysal 1.

Rekursiw ýüzlenmäni ulanyp $k!$ -y hasplamak üçin kömekçi programma düzmelili. FUNCTION FAKTOR (K: INTEGER): INTEGER;
BEGIN.

IF K=1 THEN FAKTOR:=1 ELSE
FAKTOR:=FAKTOR (K-1) *K END;
Programmanyň islendik ýerinde $n!$ hasaplamak üçin
nfaktor:= fakt (n);
operatorly ýazmak ýeterlikdir.

§4. Kömekçi programmalary ulanmaklyga degişli mysallary çözmek

Ýokarda bellenilip geçilişi ýaly programmada funksiýalary ulanmak programmanyň algoritmindäki käbir bölek meseleleri aýratynlykda çözmäge, olary funksiýalar we proseduralar görnüşinde kesgitlemäge, programmanyň operatorlar bölümünde bolsa olara ýüzlenmäge mümkinçilik

berýär. Netijede programmanyň esasy algoritmi has düşnükli bolup galýar.

Mysal 1. n natural san berlen. n-e çenli ýönekeý sanlary çap etmeli.

Çözülişi.

```
Uses crt;
Var
n,i:integer;
Function Barla(m:integer):Boolean;
Var
b:boolean;
i,k:integer;
begin
b:=true;
for i:=2 to m-1 do
if m mod i =0 then b:=False;
barla:=b;
end;
begin
Begin
clrscr;
write('n=');
for i:=1 to n do
if barla(i) then write(i,' ');
end.
```

Mysal 2. Sh $x=e^x-e^{-x}/2$ prosedurany ulanyp $z=Sh^2a+Sh(a-b)/Sha+SQR(a^2-b^2)$

hasaplamak üçin programma düzmelili.

Çözülişi.

```
PROGRAM PROSEDUR;
USES CRT;
VAR A,B, Z,T1, T2, T3:REAL;
PROCEDURE SH (X: REAL; VAR R: REAL
);
```

```

BEGIN
    R:= (EXP(X) – EXP(-X))/2.0
END;
BEGIN CLRSCR;
    WRITE('A we B bahasyny giriz ');
    READ (A, B);
    SH (A, T1);
    SH ( A-B,T2);
    SH(A*A-B*B,T3);
    Z:=(T1*T1+T2)/(T1+SQRT(T3));
    WRITE('Z=',Z:12:4);
END.

```

Eger prosedurada we baş programmada şol bir atly parametrler ulanylyan bolsalar, onda prosedura ygylan edilende formal parametrleri görkezmek hökman däl.

Mysal 3. Goý tekizlikde N-sany nokat özleriniň gönüburçly koordinatalary bilen berlen bolsun:

$$(X_i, Y_i), X_i > 0, i=1, N$$

Bu nokatlaryň polýar koordinatalaryny kesgitlemegi parametrsiz görnüşde ýazmaly:

$$\begin{aligned}
 R &= (x^2 + y^2); \\
 Tg(f) &= y/x; \\
 F &= \arctg(y/x).
 \end{aligned}$$

Çözlüşi.

```

PROGRAM POLKORD;
USES CRT;
VAR X,Y,R,F:REAL;
PROCEDURE POLAR;
BEGIN
    R:=SQRT(SQR(x)+SQR(y));
    F:=ARCTAN(Y/X);
END;
BEGIN
    CLRSCR; WRITELN('N='); READ(N);

```

```
FOR I:=1 TO N DO BEGIN
  WRITELN('X,Y bahasyny giriz');
  READ(x,y); POLYAR;
  WRITELN('R=',R:8:3,' F=', F:8:3);
  END;
END.
```

IKINJI BÖLÜM

PROGRAMMALAŞDYRMAKDA MASSIWLERIŇ ULANYLYŞY

§5. Matrissalaryň üstündäki amallar

Matrissalar iki ölçegli massiwlerdir. Iki ölçegli massiwiň her bir elementiniň iki sany indeksi bolýar. Iki ölçegli massiw beýan edilende her bir indeksiň üýtgeýiş aralygy aýratynlykda görkezilýär.

Mysal üçin,

Var

a:array[1..100,1..100] of real;

Massiwiň elementleri bilen işlenilende köplenç ýüze çykaýmagy mümkin bolan ýazgylara seredeliň. Onuň üçin ilki 1 sany massiw we 3 sany kömekçi üýtgeýän ululyk girizeliň:

VAR

Y : ARRAY[1..15,1..20] OF INTEGER;

I,J,K : INTEGER;

- 1) Y iki ölçegli massiwi nullamak:

FOR I:=1 TO 15 DO

FOR J:=1 TO 20 DO Y[I,J]:=0;

- 2) Massiwiň elementlerini girizmek:

TURBO-PASCAL algoritmik dilinde massiwiň ähli elementlerini birbada girizip ýa-da çapa çykarmaly bolýar.

Kompýuter-de köplenç massiwiň elementlerini klawiaturadan girizilýär. Meselem:

FOR I:=1 TO 15 DO

FOR J:=1 TO 20 DO READLN(Y[I,J]);

Bu ýerde Y iki ölçegli massiwiň elementlerine baha girizilýär; READLN operatorynyň ulanylýanlygy sebäpli her setirden bir baha girizilýär.

- 3) Massiwiň elementlerini çapa çykarmak:

Massiwiň elementleriniň bahalaryny çapa çykarmak hem edil ýokardaka meňzeşlikde amala aşrylýar: diňe READ, READLN operatorlarynyň ornuna WRITE, Writeln operatorlary ulanylýar. Meselem:

```
FOR I:=1 TO 15 DO
```

```
  FOR J:=1 TO 20 DO Writeln(Y[I,J]);
```

Bu ýagdaýda her setirde bir baha çap edilýär. Matrisany setirme-setir çap etmek üçin, programmany aşakdaky ýaly özgertmeli:

```
FOR I:=1 TO 15 DO
```

```
  BEGIN  Writeln;
```

```
    FOR J:=1 TO 20 DO Writeln(Y[I,J]);
```

```
  END;
```

Kä ýagdaýlarda massiwiň elementleriniň arasyndan haýsy hem bolsa bir şerti kanagatlandyryýanlaryny gözlemeli bolýar. Mysal üçin, goý $A(4 \times 5)$ gönüburçly matrisa berlen bolsun. Bu matrisanyň položitel elementleriniň jemini hasaplamagyň programmasyny düzmelili:

```
PROGRAM JEM1;
```

```
USES CRT;
```

```
CONST
```

```
  IMAX=4; JMAX=5;
```

```
TYPE
```

```
  MATR=ARRAY[1..IMAX,1..JMAX] OF REAL;
```

```
VAR
```

```
  A : MATR; JEM : REAL;
```

```
  I,J : INTEGER;
```

```
BEGIN
```

```
  CLRSCR;
```

```
  JEM:=0;
```

```
  FOR I:=1 TO I MAX DO
```

```
    FOR J:=1 TO J MAX DO
```



```

BEGIN
READLN(A[I,J]);
IF A[I,J]>0 THEN JEM:=JEM+A[I,J];
END;
WRITE('JEM=',JEM:8:3);
END.

```

Matrissalary goşmak we aýyrmak elementme-element amala aşyrylýar.

Aşakdaky programmalarda berlen $n \times m$ ölçegli matrissalar girizilýär we deňişlilikde olaryň jemi hem-de tapawudy tapylýar.

```

Program matrissalaryn_jemi;
const
n=10;  m=5;
Var
a,b,c:array[1..n, 1..m] of real;
i,j:integer;
Begin
writeln(n,'*',m,' ölçegli A matrissany girizin');
for i:=1 to n do
for j:=1 to m do read(a[i,j]);
writeln(n,'*',m,' ölçegli B matrissany girizin');
for i:=1 to n do
for j:=1 to m do read(b[i,j]);
for i:=1 to n do
for j:=1 to m do c[i,j]:=a[i,j]+b[i,j];
writeln('A we B matrissalaryň jemi');
for i:=1 to n do begin writeln;
for j:=1 to m do write(c[i,j]:4:1);
end;
end.

```

```

Program matrissalaryn_tapawudy;
const

```

```

n=10; m=5;
Var
a,b,c:array[1..n, 1..m] of real;
i,j:integer;
Begin
writeln(n,'*',m,' ölçegli A matrissany girizin');
for i:=1 to n do
for j:=1 to m do read(a[i,j]);
writeln(n,'*',m,' ölçegli B matrissany girizin');
for i:=1 to n do
for j:=1 to m do read(b[i,j]);
for i:=1 to n do
for j:=1 to m do c[i,j]:=a[i,j]-b[i,j];
writeln('A we B matrissalaryň jemi');
for i:=1 to n do begin writeln;
for j:=1 to m do write(c[i,j]:4:1);
end;
end.

```

Matrissany sana köpeltmek üçin onuň ähli elementlerini şol sana köpeltmek ýeterlikdir. Aşakdaky programmalarla berlen $n*m$ ölçegli A matrisa we k san girizilýär hem-de A matrissanyň k sana köpeltmek hasyly tapylýar.

```

Program matrissany_sana_köpeltmek;
const
n=10; m=5;
Var
a,b:array[1..n, 1..m] of real;
i,j:integer;
k:real;
Begin
write('k sany girizin'); read(k);
writeln(n,'*',m,' ölçegli A matrissany giriziň');
for i:=1 to n do
for j:=1 to m do read(a[i,j]);

```

```

for i:=1 to n do
for j:=1 to m do b[i,j]:=k*a[i,j];
writeln('A matrissanyň ', k, ' sana köpeltmek hasyly');
for i:=1 to n do begin writeln;
for j:=1 to m do write(b[i,j]:4:1);
end;
end.

```

Matrissalar köpeldilende birinji matrissanyň sütünleriniň sany ikinji matrissanyň setirleriniň sanyna deň bolmagy zerurdyr. Birinji A matrissanyň i-nji setiriniň elementleri ikinji B matrissanyň degişli j-nji sütüniniň degişli elementlerine köpeldilýär we alnan köpeltmek hasyllaryň jemi tapylýar. Alnan jem täze alynýan C matrissanyň i-nji setiriniň we j-nji sütüniniň kesişýän ýerinde ýazylýar. Matrissalaryň köpeltmek hasylyny hasaplamagyň programmasy aşakdaky görnüşde bolýar(ýönekeýlik üçin A we B matrissalar 3×3 ölçegli diýip hasap edilýär):

```

{Matrissalaryn kopeltmek hasyly}
uses crt;
const n=3;
Var
a,b,c:array[1..n,1..n] of integer;
i1,j1,i,j:integer;
begin
clrscr;
writeln('A(3x3 matrisany girizin)');
for i:=1 to n do
for j:=1 to n do
read(a[i,j]);
writeln('B(3x3 matrisany girizin)');
for i:=1 to n do
for j:=1 to n do
read(b[i,j]);
writeln('C matrissa ');
for i:=1 to n do

```

```

    for j:=1 to n do
    for i1:=1 to n do c[i,j]:=c[i,j]+a[i,i1]*b[i1,j];
for i:=1 to n do begin writeln;
    for j:=1 to n do
    write(c[i,j], ' ');
    end;
    readkey;
end.

```

§6. Çyzykly deňlemeler sistemasyny çözmekde ulanylýan Gausyň usuluny programmalaşdyrmak.

Matematikadan belli bolan Gausyň yzygider ýok etmek usulynyň Paskal dilindäki programmasy aşakdaky görnüşde bolýar:

```

Uses crt;
const R=10; R1=11;
Var
    a:array[1..R,1..R1] of real; {Koeffisiyentlerden
düzülen matrissa}
    x:array[1..R] of real;
    maxabs, v:real;
    i,j,l,k,n:integer;
BEGIN clrscr;
write('Sistemadaky denlemelerin sanyny girizin! ');read(n);
    write('Sistemadaky denlemelerin koeffisientlerini
we');
        writeln(' azat clenleri setirler boyunca girizin');
        for i:=1 to n do
            for j:=1 to n+1 do read(a[i,j]);
        for i:=1 to n do
            begin maxabs:=abs(a[i,j]); k:=i;
            for l:=i+1 to n do
                if abs(a[l,i])>maxabs then
                    begin maxabs:=abs(a[l,i]);

```

```

        k:=l; end;
        if k<>i then
        for j:=i to n+1 do
            begin v:=a[i,j]; a[i,j]:=a[k,j];
                a[k,j]:=v; end;
        v:=a[i,i];
        for j:=i to n+1 do a[i,j]:=a[i,j]/v;
for l:=i+1 to n do
    begin v:=a[l,i];
        for j:=i+1 to n+1 do
a[l,j]:=a[l,j]-a[i,j]*v;
end; end;
x[n]:=a[n,n+1];
for i:=n-1 downto 1 do
begin x[i]:=a[i,n+1];
for j:=i+1 to n do
x[i]:=x[i]-a[i,j]*x[j];
end;
for i:=1 to n do writeln('x[' ,i ,']=' ,x[i]:5:2);
readkey;
end.

```

Bu programmanyň kömegi bilen aşakdaky deňlemeler sistemasyňy çözmeli:

$$2x_1 - 4x_2 + 4x_3 = 6,$$

$$-5x_1 + 12x_2 - 14x_3 = -35,$$

$$6x_1 - 7x_2 + 5x_3 = 10.$$

$$\text{Jogap } x_1=11, \quad x_2=18, \quad x_3=14.$$

§7. Kesgitleýjileri hasaplamagyň programmasyny düzmek.

Kesgitleýjileri hasaplamakda Gaussyň yzygider ýok etmek usulyny ulanyp ol kesgitleýjini elementar özgertmeleriň kömegi bilen üçburçly görnüşe getirýäris. Soňra bolsa alnan

kesgitleýjiniň baş diagonalyndaky elementleriň köpeltmek hasylyny tapýarys.

Islendik tertipli kesgitleýjini hasaplamagyň programmasy aşakdaky görnüşde bolýar:

```
Uses crt;
const R=10; R1=11;
Var a:array[1..R,1..R1] of real; {Koeffisiyentlerden
duzulen matrissa}
    x:array[1..R] of real;
    S,maxabs, v:real;
    i,j,l,k,n:integer;
BEGIN clrscr;
s:=1;
write('Kesgitleýjiniň olcegini girizin! ');read(n);
    write('Kesgitleýjiniň elementlerini');
    writeln(' setirler boyunca girizin');
        for i:=1 to n do
            for j:=1 to n do read(a[i,j]);
        for i:=1 to n do
            begin maxabs:=abs(a[i,j]); k:=i;
            for l:=i+1 to n do
                if abs(a[l,i])>maxabs then
                    begin maxabs:=abs(a[l,i]);
                    k:=l; end;
            if k<>i then
                for j:=i to n do
                    begin v:=a[i,j]; a[i,j]:=a[k,j];
                    a[k,j]:=v; end;
            v:=a[i,i]; s:=s*v;
            for j:=i to n do a[i,j]:=a[i,j]/v;
        for l:=i+1 to n do
            begin v:=a[l,i];
            for j:=i+1 to n do
                a[l,j]:=a[l,j]-a[i,j]*v;
        end ; end; writeln('S=',S); readkey; end.
```

Bellik:

Bu programmanyň dogry işlemegi üçin berlen kesgitleýjiden özgerdiliş alynýan kesgitleýjiniň baş diagonalynda nul sanyň alynmagy mümkin bolmaly däl. Eger tersine bolsa, onda programma ýerine ýetirilýän wagtynda nula bölmek alynar we programma işlemegini bes eder.

§8. Ters matrissany hasaplamagyň programmasyny düzmek.

Ýokary matematikadan belli bolşy ýaly berlen kwadrat A matrissanyň esasy kesgitleýjisi nuldan tapawutly bolan ýagdaýynda onuň ters A^{-1} matrissasyny tapmak mümkin. A matrissanyň ters matrissasy

$$A \cdot A^{-1} = E \quad (1)$$

deňlikden tapylýar, bu ýerde E birlik matrissadyr. (1) matrissa deňligi giňeldilip ýazylanda n^2 sany näbellili n^2 sany çyzykly deňlemelerden ybarat bolan deňlemeler sistemasy alynýar. Şol sistemada A^{-1} matrissanyň elementleri näbelli ululyklar bolar.

Aşakdaky programma 2 ölçegli kwadrat matrissanyň ters matrissasyny tapmaga mümkinçilik berýär:

```

Program tersmatrissa2;
Uses crt;
Var a,b:array[1..2,1..2] of real;
i,j:integer;
e,f:real;
Begin clrscr;
writeln('Berlen A(2*2) matrissanyň elementlerini girizin!');
for i:=1 to 2 do
  for j:=1 to 2 do read(a[i,j]);
e:=a[2,1]*a[1,2];   f:=a[1,1];   e:=a[2,2]*f-e;
a[1,1]:=a[2,2]/e;
a[2,2]:=f/e;   a[2,1]:=-a[2,1]/e;   a[1,2]:=-a[1,2]/e;
writeln('Ters matrissa');
for i:=1 to 2 do begin writeln;
```

```
for j:=1 to 2 do write(a[i,j]:4:0);  
end;  
readkey;  
end.
```

§9. Massiwleri tertiplemekdäki ulanylýan ýönekeýje usullar.

Biziň her birimiz rusça-türkmençe ýa-da iňlisçe-rusça sözlükleri ulanyp görendiris. Sözlüklerde sözleriň elipbiý boýunça ýerleşdirilmegi ondan gerekli sözümi çalt tapmaga mümkinçilik berýär. Eger-de sözlükdäki sözler elipbiý boýunça tertipleşdirilmän ýerleşdirilen bolsa, onda sözlükden peýdalanmak gaty kynlaşardy we köp wagtymyzy alardy. Başga bir mysal: Aşgabat şäherindäki telefonlar barada maglumat berýän kitap. Bu kitapda maglumatlar müşderileriň telefon nomerleri ýa-da müşderileriň familiýalary boýunça ýerleşdirilýär. Bu mysallardan görnüşi ýaly durmuşda maglumatlary tertipleşdirip goýmaklyga degişli zerurlyk bardyr. Biz bu temada san ululyklary tertipleşdirmek meselesine serederis, ýöne sözleri tertipleşdirmek meselesi hem biziň seredýän meselelerimize meňzeş çözülýär. Sebäbi simwollary deňeşdirmegi olaryň kodlaryny deňeşdirmeklige syrykdyryp bolýar. Sanlary tertipleşdirmek diýip olary artýan ýa-da kemelýän tertipde ýerleşdirmeklige düşüneris.

Maglumatlary tertipleşdirmegiň usullary iki topara bölünýär: içki we daşky. İçki usullarda operatiw huşdaky maglumatlar tertipleşdirilýär, daşky usullarda bolsa faýllardaky ýerleşen maglumatlar tertipleşdirilýär. Biz tertipleşdirmegiň içki usullaryna seredip geçeris.

Düwmejik usuly

Massiwleri tertipleşdirmegiň köp sanly algoritmleri bardyr. Düwmejik usulynda massiwiň birinji elementinden başlap iň soňky elementine çenli zygyder goňşy elementler

deňeşdirilýär. Eger nobatdaky deňeşdirmedäki iki element tertipleşdirilmedik bolsa, onda olaryň orny çalýyrylýar. Netijede iň uly(kiçi) element iň soňky orna barar. Şeýle deňeşdirmeklige jemi $n-1$ gezek gaýtalamaly. Netijede massiw tertipleşer.

Düwmejik usulynyň programmasy aşakdaky ýaly bolýar:

```
{Düwmejik usuly}
Uses crt;
Var
a:array[1..20] of integer;
c,i,j:integer;
begin
  clrscr;
  randomize;
  for i:=1 to 20 do read(a[i]);
  writeln('Berlen massiw');
  for i:=1 to 20 do write(a[i], ' ');
  readkey;
  for i:=1 to 19 do
    for j:=1 to 19 do
      if a[j]>a[j+1] then begin
        c:=a[j];
        a[j]:=a[j+1];
        a[j+1]:=c; end;
  writeln;
  writeln('Tertipleshdirilen massiw');
  for i:=1 to 20 do write(a[i], ' ');
  readkey;
end.
```

Uly elementler usuly

Bu usulda ilki birinji elementden başlap massiwiň elementleriniň arasynda iň kiçisi tapylýar. Ol elementiň ýeri

birinji element bilen çalşyrylýar. Soň ikinji elemntden başlap ýokarky iş gaýtalanýar. Netijede n-1 gezekden soň massiw tertipleşer. Uly elementler usulyňyň programmasy aşakdaky görnüşdedir:

```
Uses crt;
Var
a:array[1..20] of integer;
min,imin,i,j:integer;
begin
clrscr;
randomize;
for i:=1 to 20 do a[i]:=random(90)+10;
writeln('Berlen massiw');
for i:=1 to 20 do write(a[i], ' ');
readkey;
for i:=1 to 19 do begin min:=a[i];imin:=i;
for j:=i+1 to 20 do
if a[j]<min then begin min:=a[j];imin:=j;end;
a[imin]:=a[i];
a[i]:=min; end; writeln;
writeln('Tertipleşdirilen massiw');
for i:=1 to 20 do write(a[i], ' ');
readkey;
end.
```

Goşmak bilen tertipleşdirmek

Tertiplemegiň bu usulynda elementleriň yzygiderligi iki topara bölünýär: 1-nji bölüm tertipleşdirilen bölüm; 2-nji bölüm entäk tertipleşdirilmedik bölüm. Ilki başda 1-nji topara diňe massiwiň birinji elementi girýär. Soň 2-nji topardan 1-nji topara elementler yzly-yzyna geçirilip başlanýar. 1-nji topara geçirilýän her bir element öz durmaly ýerine goýulýar. Netijede 1-nji topar elmydama tertipleşen ýagdaýda bolýar. 2-nji topardaky iň soňky element 1-nji topara goşulmagy bilen

algoritmi tamamlanýar. Tertipleşdirmegiň bu usuly kart oýnunda oýunçynyň elindäki kartlary tertipleşdirşine meňzeş amala aşyrylýar.

Bu usulyň programmasy aşakdaky görnüşde bolýar.

```
Uses crt;
Label 1;
Var
a:array[1..100] of integer;
n,c,i,j,k,t:integer;
Begin
clrscr;
randomize;
writeln('A massiw elementlerinin sanyny giriz'); read(n);
for i:=1 to n do a[i]:=random(100);
writeln;
for i:=1 to n do write(a[i], ' '); writeln(' Berlen massiw');
writeln;
for i:=2 to n do
begin
c:=a[i]; k:=1;
for j:=1 to i-1 do
if c>a[j] then k:=k+1 else goto 1;
1:
for t:=i downto k do
a[t]:=a[t-1];
a[k]:=c;
for t:=1 to n do write(a[t], ' '); write(' ',i-1,'-nji adim ');
writeln;
end;
writeln;
writeln('Alnan massiw');
for i:=1 to n do write(a[i], ' ');
readkey;
end.
```

Çalt tertiplemek usuly

1962-nji ýylda Çalz Hoar tarapyndan massiwleri tertipleşdirmeginiň aşakdaky algoritmi tekliپ edildi we ol massiwleri çalt tertiplemegiň usuly diýip atlandyryldy:

1. Massiwi girizmeli;
2. Massiwiň bir elementini saýlap almaly(ortasyndan ýa-da başga hili) we massiwi iki bölege bölmeli. Şol bahadan kiçi elementleri birinji bölege, ulylary bolsa ikinji bölege geçirmeli.
3. Her bir bölek üçin, alnan bölekde birden köp element bar bolsa, ikinji ädimi gaýtalamaly.
4. Alnan massiwi çap etmeli.

Çalt tertiplemek usulynyň programmasy aşakda getirilýär:

```
Uses crt;
```

```
Type
```

```
AT=array[1..100] of integer;
```

```
Var
```

```
a:AT;
```

```
n,i:integer;
```

```
procedure exchange(var a,b:integer);
```

```
var c:integer;
```

```
begin
```

```
c:=a;
```

```
a:=b;
```

```
b:=c;
```

```
end;
```

```
procedure QuickSort(Var a:at; first,last:integer);
```

```
Var
```

```
v:integer;
```

```
left,right:integer;
```

```
begin
```

```
left :=first;
```

```
right:=last;
```

```

v:=a[(left+right) div 2];
while left<=right do begin
  while a[left]<v do inc(left);
  while a[right]>v do dec(right);
  if left<= right then begin
    exchange(a[left],a[right]);
    inc(left); dec(right);end;
  end;
  { left>right }
  if first<right then
    quickSort(a,first,right);
  if left<last then
    quickSort(a,left,last);
end;
begin
clrscr;
write('n='); read(n);
randomize;
for i:=1 to n do a[i]:=random(100);
writeln('Berlen massiw');
for i:=1 to n do write(a[i], ' ');
QuickSort(a,1,n);
writeln;
writeln('Tertipleşdirilen massiw');
for i:=1 to n do write(a[i], ' ');
readkey;
end.

```

MATEMATIKI MESELELERI PROGRAMMALAŞDYRMAK

§10. Deňlemeleri çözmegiň ýakynlaşan usullary

Ýokary matematikadan belli bolşy ýaly ähli görnüşli deňlemeleri takyk çözmek üçin umumy usul ýa-da formula ýokdur. Şol sebäpli deňlemeleri çözmegiň ýakynlaşan usullary ulanylýar. Olaryň käbirine seredip geçeliň.

Iterasiýa usuly

Goý

$$f(x)=0 \quad (1)$$

deňlemäniň köküniň ýakynlaşan bahasyny tapmak talap edilýän bolsun. (1) deňlemäni

$$x=\varphi(x) \quad (2)$$

görnüşde ýazmak mümkin bolsun, bu ýerde $\varphi(x)$ funksiýasy berlen $[a,b]$ aralykda differensirlenýän we şol aralykda $|\varphi(x)| < 1$ bolsun. Iterasiýalar usuly bilen köküň ýakynlaşan bahasyny tapmak üçin berlen $[a,b]$ aralykdan köküň başlangyç x_0 bahasy alynýar. Ýakynlaşma

$$x_n=\varphi(x_{n-1}) \quad (3)$$

formula boýunça gurulýar. Hasaplama $|x_n - x_{n-1}| < \varepsilon$ şert ýerine ýetýänçä dowam etdirilýär, bu ýerde ε – berlen takyklyk. Iterasiýa usulynyň programmasy aşakdaky görnüşde bolýar:

Uses Crt;

Var

x0,x1,x,eps:real;

Function F(x:Real):Real;

begin

F:=x*x*x-3*x-5

```

end;
Begin
clrscr;
write('x0=');
read(x0);
Write('Takyklyk=');
Read(eps);
x1=F(x0);
if abs(x1-x0)>eps then
begin x0:=x1; x1=F(x0); end ;
x=x1;
writeln('x=',x);
end.

```

Kesimi ýarpa bölmek usuly

Goý bize görnüşli deňleme berlen bolsun, şol ýerde $f(x)$ – berlen $[a,b]$ aralykda üznüksiz funksiýa bolsun. Goý (1) deňlemäniň $[a,b]$ aralykda bir we diňe bir ýönekeý köki bar bolsun. Kesimi ýarpa bölmek usuly aşakdaky teorema esaslanýar:

Eger $[a,b]$ aralykda (1) deňlemäniň bir köki bar bolsa, onda

$$f(a)*f(b) \leq 0 \quad (2)$$

şert ýerine ýeter, hususanda, $f(a)*f(b) < 0$ şert ýerine ýetýän bolsa, onda (1) deňlemäniň köki $[a,b]$ aralygyň içinde, a egerde $f(a)*f(b) = 0$ bolsa, onda deňlemäniň köki $[a,b]$ kesimiň uçlarynda ýerleşýär.

Eger kök $[a,b]$ aralygyň içinde ýatýan bolsa, onda $c=(a+b)/2$ nokat ($[a,b]$ kesimiň ortasy) tapylýar. Soňra $f(c)=0$ şert barlanylýar. Eger ol ýerine ýetýän bolsa, onda $x=c$ baha deňlemäniň köki bolýar, a eger $f(c) \neq 0$ şert ýerine ýetmeýän bolsa, onda $f(a)*f(c) < 0$ şert barlanylýar. Eger bu şert ýerine ýetýän bolsa, onda gözlenilýän kök $[a,c]$ aralykda, tersine bolsa kök $[c,b]$ aralykda ýatýar. Şol alynan aralyklar $[a,b]$ aralyk derejine alynýar. Soň ýenede ýokardaky işler gaýtalanýar. Gaýtalamak gözlenilýän kök takyk tapylanda ýa-da $|b-a| < \epsilon$

şert ýerine ýetýänçä dowam edilýär. Bu ýerde eps berlen takyklyk.

Mysal. $x^3+3x-5=0$ deňlemäniň [1,2] aralykda ýatýan köküni tapmagyň programmasyny düzmeli.

Meseläni çözmegiň algoritmi aşakdaky görnüşde bolýar:

```
Uses Crt;
Label 1,2;
Var
x,a,b,c,eps:real;
Function F(x:Real):Real;
begin
F:=x*x*x-3*x-5
end;
Begin
clrscr;
write('a,b=');
read(a,b); { a=1,b=2 berilmeli }
Write('Takyklyk=');
Read(eps);
if abs(f(a))<=1E-10 then
    begin x:=a; goto 2; end
    else if abs(f(b))<=1E-10 then
        begin x=b; goto 2; end;
1: c=(a+b)/2;
if abs(f(c))<=1E-10 then begin x:=c; goto 2; end;
if F(a)*F(c)<0 then b:=c else a:=c;
if abs(b-a)<= eps then x:=(a+b)/2 else goto 1;
2: writeln('x=',x); end.
```

Hordalar usuly

Bu usulda $f(x)=0$ deňlemäniň ýeke täk bir köküni özünde saklaýan $[a,b]$ kesim tapylýar. Goý şol aralykda $f(x)$ üznüksiz funksiýa bolsun. $y=f(x)$ funksiýanyň grafiginiň uçlary bolan

$A(a, f(a))$ we $B(b, f(b))$ nokatlaryň üstünden horda geçirýäris. Ol hordanyň deňlemesi

$$\frac{x-a}{b-a} = \frac{y-f(a)}{f(b)-f(a)} \quad \text{görnüşde bolýar. Bu hordanyň ox}$$

oky bilen kesişme nokadyny C bilen belgiläliň. C nokadyň koordinatalary

$$x_C = a - (b-a) \frac{f(a)}{f(b)-f(a)} \quad (3)$$

we $y=0$ bolar. Berlen $[a,b]$ aralygyň deregine $[a,c]$ ýa-da $[c,b]$ kesimleriň biri alynýar. Has takygy, eger gözlenilýän kök $[a,c]$ kesimde ýatýan bolsa, onda $[a,b]$ kesimiň deregine $[a,c]$ kesimi almaly ($b:=c$), bolmasa $[c,b]$ kesimi almaly ($a:=c$). Bu gaýtalamak prosesi gözlenilýän kökiň bahasyna berlen takyklykda ýakynlaşylýança dowam etdirilýär. Köplenç halatda hordalar usuly galtaşýanlar usuly bilen utgaşdyrylyp ulanylýar.

Galtaşýanlar usuly

Bu usul bilen $f(x)=0$ deňleme çözülende ilkibaşda deňlemäniň ýeke täk bir köküni özünde saklaýan $[a,b]$ kesim tapylýar. Eger $[a,b]$ kesimde $f(x)$ funksiýa ösýän bolsa we onuň grafigi güberçekligi aşaklygyna ýüzlenen (ýagny $f'(x)>0$, $f''(x)>0$) bolsa, onda grafigiň soňky nokady bolan $(b, f(b))$ nokatdan $y=f(x)$ funksiýanyň grafigine galtaşýan çyzyk geçirilýär we onuň ox oky bilen kesişme nokady tapylýar. Ol nokadyň absissasy

$$x=b - b/f(b) \quad (4)$$

formula boýunça tapylýar. Eger $[a,b]$ kesimde $f(x)$ funksiýa kemelýän bolsa we onuň grafigi güberçekligi aşaklygyna ýüzlenen (ýagny $f'(x)<0$, $f''(x)>0$) bolsa, onda galtaşýan grafigiň başky $(a, f(a))$ nokadyndan geçirilýär we (4) formula derek

$$x=a - a/f(a) \quad (5)$$

formula alynýar. Deňlemeleri çözmegiň galtaşýanlar usuly hordalar usuly bilen utgaşdyrylyp ulanylýar. Has takygy deňlemäniň köküne çep tarapdan bir usul, beýleki tarapdan bolsa beýleki usul bilen ýakynlaşylýar.

Aşakda berlen $f(x)$ deňlemäniň ýeke täk kökünü tapmak üçin programma getirilýär.

```
Uses crt;
label 1,2;
Var
x,a,b,eps:real;
n:integer;
{ ----- }
Function F(x:real):real; { f(x)-Berlen denlemanın chep tarapy }
Begin
f:=x*x-1;
end;
{ ----- }
Function F1(x:real):real; { f(x) -in onumi }
Begin
f1:=2*x;
end;
{ ----- }
Function galtashyan(a:real):real;
Begin
galtashyan:=a-a/f1(a);
end;
{ ----- }
Function Horda(a,b:real):real;
Begin
Horda:=(b-a)*f(a)/(f(b)-f(a));
end;
{ ----- }
Begin
clrscr;
writeln('a,b,eps,Yagday =');read(a,b,eps,n);
```

```

1: if abs(f(a))<0.00000000001 then begin x:=a; goto 2; end;
if abs(f(b))<0.00000000001 then begin x:=b; goto 2; end;
if n=1 then begin a:=horda(a,b);
b:=galtashyan(b); end else begin b:=horda(a,b);
a:=galtashyan(a);end;
if abs(b-a) <eps then begin x:=(a+b)/2; goto 2;end Else goto 1;
2:write('x=',x);
end .

```

§11. Kesgitli integraly hasaplamak

Kesgitli integralyň takyk bahasyny tapmak üçin Nýuton-Leybnisiň formulasy ulanylýar. Bu formulany ulanmak üçin integral aşagyndaky funksiýanyň asyl funksiýasyny tapmak gerek bolýar. Emma käbir funksiýalar üçin asyl funksiýany tapmak örän kyn, käbir ýagdaýlarda bolsa düýbünden mümkin däldir. Şol sebäpli kesgitli integrallary tapmak üçin ýakynlaşan usullar ulanylýar. Biz şol usullaryň käbirine garap geçeris we degişli programmalary düzeris.

Göniburçluklar usuly

Göniburçluklar usulynda aşakdaky formula ulanylýar:

$$\int_a^b f(x)dx \approx h(f(a) + f(x_1) + f(x_2) + \dots + f(x_{n-1})),$$

bu ýerde n - berlen $[a,b]$ aralygy böleklere bölmeginiň sany, $h=(b-a)/n$, $x_1=a+h$; $x_i=x_{i-1}+h$, $i=2,\dots, n-1$. Bu usuly programmirlänimizde integralyň bahasyny berlen $[a,b]$ aralygy n we $2*n$ böleklere bölüp iki usul bilen hasaplaýarys. Eger olýň bahalarynyň bir-birinden tapawudy berlen ε takyklykdan kiçi bolsa, onda hasaplama tamamlanýar we integralyň $2*n$ –e degişli bahasy jogap hökmünde alynýar, bolmasa n –iň ýerine $2*n$ alynýar we ýokarky proses gaýtalanýar.

Göniburçluklar usulynyň programması

$\int_a^b (\sin x + x^2) dx$ integralyň mysalynda aşakdaky görnüşde

bolýar:

Uses Crt;

Var

a,b,S,S1,h,eps:real;

n,i:Integer;

Function F(x:real):Real;

begin

f:=sin(x)+sqr(x)

end;

Begin

ClrScr;

Write('a,b,eps=');

Read(a,b,eps);

1: S:=0;

n:=100;

h:=(b-a)/n;

for i:=1 to n do S:=S+f(a+(i-1)*h);

S:=S*h;

S1:=0;

n:=2*n;

h:=(b-a)/n;

for i:=1 to n do S1:=S1+f(a+(i-1)*h);

S1:=S1*h;

If abs(S-S1)<=eps then

writeln('Integral=',S1)

else

begin

n:=n*2;

goto 1;

end;

end.

Trapesiýa usuly

Trapesiýalar usulynda aşakdaky formula ulanylýar:

$$\int_a^b f(x)dx \approx h((f(a) + f(b))/2 + f(x_1) + f(x_2) + \dots + f(x_{n-1})),$$

bu ýerde n - berlen $[a,b]$ aralygy böleklere bölmeginiň sany, $h=(b-a)/n$, $x_1=a+h$; $x_i=x_{i-1}+h$, $i=2,\dots, n-1$. Bu usuly programmirlänimizde integralyň bahasyny berlen $[a,b]$ aralygy n we $2*n$ böleklere bölüp iki usul bilen hasaplaýarys. Eger olýň bahalarynyň bir-birinden tapawudy berlen ε takyklykdan kiçi bolsa, onda hasaplama tamamlanýar we integralyň $2*n$ –e degişli bahasy jogap hökmünde alynýar, bolmasa n –iň ýerine $2*n$ alynýar we ýokarky proses gaýtalanýar.

Trapesiýalar usulyňyň programmasy $\int_a^b (\sin^2 x + x) dx$

integralyň mysalynda aşakdaky görnüşde bolýar:

Uses Crt;

Var

a,b,S,S1,h,eps:real;

n,i:Integer;

Function F(x:real):Real;

begin

f:=sqr(sin(x))+x;

end;

Begin

ClrScr;

Write('a,b,eps=');

Read(a,b,eps);

1: S:=(f(a)+f(b))/2;

n:=100;

h:=(b-a)/n;

for i:=1 to n-1 do S:=S+f(a+(i-1)*h);

```

S:=S*h;
S1:=(f(a)+f(b))/2;
n:=2*n;
h:=(b-a)/n;
for i:=1 to n do S1:=S1+f(a+(i-1)*h);
S1:=S1*h;
If abs(S-S1)<=eps then
    writeln('Integral=',S1)
    else begin n:=n*2; goto 1; end;
end.

```

Simpsonyň usuly

Simpsonyň usulynda aşakdaky formula ulanylýar:

$$\int_a^b f(x)dx \approx h/3(f(a) + f(b) + 2(f(x_1) + f(x_3) + \dots + f(x_{2n-1})) + 4(f(x_2) + \dots + f(x_{2n-2})))$$

bu ýerde n - jübüt san, berlen $[a,b]$ aralygy böleklere bölmegiň sany, $h=(b-a)/n$, $x_1=a+h$; $x_i=x_{i-1}+h$, $i=2,\dots, n-1$. Bu usuly programmirlänimizde integralyň bahasyny berlen $[a,b]$ aralygy n we $2*n$ böleklere bölüp iki usul bilen hasaplaýarys. Eger olýň bahalarynyň bir-birinden tapawudy berlen ε takyklykdan kiçi bolsa, onda hasaplama tamamlanýar we integralyň $2*n$ –e degişli bahasy jogap hökmünde alynýar, bolmasa n –iň ýerine $2*n$ alynýar we ýokarky proses gaýtalanýar.

Simpsonyň usulyňyň programmasy $\int_a^b (\sin^2 x + x) dx$

integralyň mysalynda aşakdaky görnüşde bolýar:

Uses Crt;

Var

a,b,S,S1,h,eps:real;

```

n,i:Integer;
Function F(x:real):Real;
begin
f:=sqr(sin(x))+x;
end;
Function IntSim(a,b,n:real):real;
Var i:integer;
S0,S1, S2,S:real;
begin
  S0:=f(a)+f(b);
h:=(b-a)/n; S1:=0; S2:=0;
for i:=1 to n-1 do
if odd(i) then S1:=S1+f(a+i*h) else S2:=S2+f(a+i*h)
IntSim:=h/3*(S0+2*S1+4*S2);
end;
Begin ClrScr;
Write('a,b,eps=');
Read(a,b,eps);
  n:=100;
1: S:=IntSim(n);
S1:=IntSim(2*n);
If abs(S-S1)<=eps then
      writeln('Integral=',S1)
      else begin n:=n*2; goto 1; end;
end.

```

DÖRDÜNJI BÖLÜM

SENENAMALAR BILEN BALANYŞYKLY MESELELERI PROGRAMMALAŞDYRMAK

§12. Grigorýan we Ýulian kalendarlary barada umumy maglumatlar

Kalendar baradaky meseleler çözülende häzirkí döwürde ulanylýan (grigorýan) kalendarýnda biziň eramyzyň 1-nji ýylynyň Ýanwar aýynyň 1-i, hepdäniň duşenbe güni bolýandygy we ýyldaky günleriň sanynyň şol ýylyň nomerine baglydygy ulanylýar. Ýylyň nomeri 4-e bölünmeýän bolsa ol ýyl gysga ýyl(365 gün) bolýar. Ýylyň nomeriniň 4-e bölünýänleriniň arasyndan 100-e bölünip 400-e bölünmeýänleri hem gysga ýyl bolýar. Galan ýyllar uzyn ýyl bolýar.

Grigorýan kalendarý bilen birlikde Ýulian kalendarý hem ulanylýar. Ýulian kalendarý hem başlangyjyny biziň eramyzyň 1-nji ýylynyň Ýanwar aýynyň 1-inden alyp gaýdýar we ol gün hepdäniň duşenbe güni hasaplanýar. Ýöne Ýulian kalendarýnda, Grigorýan kalendarýndan tapawutlylykda, nomeri 4-e bölünýän her bir ýyl uzyn bolýar.

Aşakdaky programmada ýylyň nomeri berilýär. Şol ýylda näçe gün bardygy tapylýar.

```
uses crt;
```

```
var
```

```
    y:integer;
```

```
{ ----- }
```

```
function f(y:integer):boolean;
```

```
begin
```

```
f:=(y mod 400=0) or ((y mod 4=0) and (y mod 100<>0));
```

```
end;
```

```
{ ----- }
```



```

begin
clrscr;
write('y='); read(y);
if f(y) then write(366) else write(365);
readkey;
end.

```

§13. Baky kalendar programmasy

Güni, aýy we ýyly görkezýän a , b , c natural sanlar berlen. Şol senäniň hepdeäniň näçinji gününe düşýändigini tapmaly. Bu mesele "Baky kalendar" diýip atlandyrylýar.

Çözülişi. Programmada $p(y)$ funksiýany ulanarys. Bu funksiýa logiki tipli funksiýa bolup uzyn ýyl üçin TRUE bahany, gysga ýyl üçin bolsa FALSE bahany alar. Programmadaky a massiw yyldaky aýlarda bar bolan günleriň sanyny görkezýär. Eranyň başyndan programmada girizilýän senä çenli geçen günler sanalanda alnan jemiň 7-ä kratny bölekleri taşlanylýar, sebäbi 7 hepdedäki günleriň sanyny aňladýar.

```

uses crt;
var
  g,a,y,i,j:integer;
{ ++++++ }
function f(g,ay,y:integer):byte;
var
  a:array[1..12]of 1..31;
  i,s:longint;
{ ===== }
function p(y:integer):boolean;
begin
p:=(y mod 400=0) or ((y mod 4=0) and (y mod 100<>0));
end;

```

```

{=====}
begin
a[1]:=31; a[2]:=28; a[3]:=31; a[4]:=30; a[5]:=31; a[6]:=30;
a[7]:=31;
a[8]:=31; a[9]:=30; a[10]:=31; a[11]:=30; a[12]:=31;
if p(y) then a[2]:=29;
s:=g mod 7;
for i:=1 to ay-1 do s:=(s+a[i]) mod 7;
for i:=1 to y-1 do
if p(i) then s:=(s+366) mod 7 else s:=(s+365) mod 7;
f:=s;
end;
{+++++}
begin
clrscr;
write('g='); read(g);
write('a='); read(a);
write('y='); read(y);
write(f(g,a,y));
readkey;
end.

```

§14. Baky kalendar programmasyny senenamalar bilen baglanyşykly dürli meseleleri çözmekde ulanmak

Ýulian kalendaryndan grigorýan kalendaryna geçmek

Ýulian kalendary boýunça günü, aýy we ýyly görkezýän a , b , c natural sanlar berlen. Şol senäniň häzirki ulanylýan kalendarlarda haýsy senä düşýändigini tapmaly. Bu kalendarlary tapawut ýulian kalendarynda nomeri 4-e bölünýän her bir ýylyň uzyn ýyl bolýandygy bilen kesgitlenýär. Aşakda meseläni çözmegiň programmasy getirilýär.

```

{ Yulian to Grigorian }
uses crt;

```

```

var
  ay : array[1..12] of 1..31;
  g,a,y,g1,a1,y1 : word;
  s : longint;
{-----}
function uz(n : word) : boolean;
begin
  uz:=(n mod 400=0) or ((n mod 4=0) and (n mod 100 <> 0));
end;
{-----}
function f(g,a,y : word) : longint;
var s : longint;
    i : integer;
begin
  s:=g;
  for i:=1 to a-1 do s:=s+ay[i];
  for i:=1 to y-1 do
    if i mod 4=0 then s:=s+366 else s:=s+365;
  f:=s; end;
{-----}
procedure p(n : longint; var g,a,y : word);
var i : word;
begin
  y:=1;
  repeat
    if uz(y) then n:=n-366 else n:=n-365;
    inc(y);
  until n<365;

  a:=1;
  while n>ay[a] do
  begin
    n:=n-ay[a];
    inc(a);
  end;

```

```

    g:=n; end;
{-----}
begin clrscr;
    writeln('Yulian --> Grigorian');
    write('gun='); read(g);
    write('ay='); read(a);
    write('yyl='); read(y);
    ay[1]:=31; ay[2]:=28; ay[3]:=31; ay[4]:=30; ay[5]:=31;
ay[6]:=30;
    ay[7]:=31; ay[8]:=31; ay[9]:=30; ay[10]:=31; ay[11]:=30;
ay[12]:=31;
    if uz(y) then ay[2]:=29;

    s:=f(g,a,y);
    p(s,g1,a1,y1);
    writeln(g1,'-',a1,'-',y1);    readkey; end.

```

Okalgadaky arassaçylyk günlerini tapmak

Käbir kitaphanada her aýyň iň soňky sogap günü arassaçylyk günü geçirilýär we şol gün kitaphana okyjylara hyzmat etmeýär. Ýyly görkezýän natural san berilýär. Şol ýylda kitaphanadaky geçiriljek arassaçylyk günleriniň ählisini tapmaly.

Meseläni çözmekligiň programmasy aşakdaky görnüşdedir.

```

uses crt;
var
    ay : array[1..12] of 1..31;
    g,a,y,i,k,n : integer;
{-----}
function uz(n:word) : boolean;
begin
    uz:=(n mod 400 = 0) or ((n mod 4 = 0) and (n mod 100 <>
0));

```

```

end;
{-----}
procedure init;
begin
    ay[1]:=31; ay[2]:=28; ay[3]:=31; ay[4]:=30; ay[5]:=31;
ay[6]:=30;
    ay[7]:=31; ay[8]:=31; ay[9]:=30; ay[10]:=31; ay[11]:=30;
ay[12]:=31;
end;
{-----}
function f(g,a,y:word):longint;
var s : longint;
    i : integer;
begin
{    if uz(y) then ay[2]:=29;}
    s:=g;
    for i:=1 to a-1 do s:=s+ay[i];
    for i:=1 to y-1 do
        if uz(i) then s:=s+366 else s:=s+365;
    f:=s;
end;
{-----}
procedure giriz;
begin
    write('Year='); read(y);
    write('Day of Week (last)='); read(n);
    if uz(y) then ay[2]:=29;
end;
{-----}
begin clrscr;
    giriz;
    init;
    k:=0;
    for a:=1 to 12 do
        for g:=ay[a] downto 1 do

```

```
    if  $f(g,a,y) \bmod 7 = n$  then  
        begin  
            inc(k);  
            writeln(g, ' ', a, ' ', y);  
break;  
        end;  
    end.
```

TEKST FAÝLLARY BILEN İŞLEMEK

§15. Simwollary we setirleri gaýtadan işlemeklige degişli meseleleri programmalaşdyrmak

Simwollary gaýtadan işlemek

Turbo Paskalda simwol (char) we setir (string) tipli üýtgeýänler bilen işlemeklik mümkindir.

Char tipli baha – kompýuteriň elipbiýindäki boş bolmadyk, goşa apostrof belgisiniň içine alnan, simwoldyr. Mysal üçin, ‘ ‘, ‘A’, ‘7’, we ş.m. Simwollaryň bu adaty ýazgysyndan başga Turbo Pascal simwollary ýazmak üçin ýene-de iki görnüşi ulanýar:

1. Simwoly onuň ASCII kodynyň kömegi bilen aňlatmak. Bu ýagdaýda simwolyň kodunyň önünden # belgisi goýulýar. Mysal üçin,

97 = Chr(97) = ‘a’ (‘a’ simwol),
65 = Chr(65) = ‘A’ (‘A’ simwol),
32 = Chr(32) = ‘ ’ (boşluk, probel),
0 = Chr(0) (nolunjy simwol).

2. Kodlary 1-den 31-e çenli bolan dolandyryjy simwollar. Olary belgilemek üçin “^” belgi we latyn elipbiýindäki şol nomerli harp(1...26 aralykdaky kodlar üçin) ýa-da kömekçi belgi (27...31 aralykdaky kodlar üçin) ulanylýar. Mysal üçin,

^A = #1 = Chr(1) - kod 1,
^B = #2 = Chr(2) - kod 2,
...
^ = #26 = Chr(26) - kod 26,
^[= #27 = Chr(27) - kod 27,
...
^_ = #31 = Chr(31) - kod 31,

şol sanda ^G – ses (7), ^I – TAB (kod 9), ^J – LF (kod 10), ^M– CR (kod 13) we ş.m.

Tekstleri gaýtadan işlemek üçin STRING(setir) tipi ulanylýar. Ol öz gurluşy boýunça simwollaryň birölçegli *ARRAY[0..N] of CHAR* massiwine meňzeşdir. Setir tipli üýtgeýän ululyklar beýan edilende STRING ýa-da STRING[N] sözler ulanylýar. Bu ýerde N setiriň bolup biljek uzynlygydyr. Eger ol görkezilmedik bolsa N=255 diýip hasap edilýär.

Programmada setiriň her bir simwolyňa gös-göni ýüzlenip bolýar, Onuň üçin ilki setiriň ady, soňra bolsa dik ýaýlaryň içinde şol simwolyň setirdäki tertip nomeri ýazylýar Mysal üçin, s[5], a[23], d[n].

Setirdäki simwollaryň tertip nomeri 0-dan başlanýar. 0-njy siinwol setiriň uzynlygyny aňladýar Mysal üçin, S[0] S setiriň uzynlygyny görkezýän simwoldyr. Setiriň uzynlygynda 0-njy simwol hasaba alynmaýar, şonuň üçin setiriň ilkinji simwolyň nomeri 1 deňdir. Setiriň uzynlygynyň bitin sanda aňladylyşyny almak üçin ord(S[0]) funksiýasyny ulanmalydyr. Bu ýerde ord -funksiýasy simwolyň kodyny kesgitleýär.

Simwollary diňe deňeşdirmek we dakmak (onuň kömegi bilen başga simwola baha bermek) bolýar. Simwollar deňeşdirilende olaryň ASCII kodlary deňeşdirilýär. Eger simwollaryň kodlary deň bolsa, onda olar deň hasaplanýar, a eger bir simwolyň kody beýlekiniňkiden uly bolsa, onda kody uly simwol uly hasaplanýar:

‘R’ = ‘R’

‘r’ > ‘R’ (kod 114 > kod 82).

Deňeşdirmek amallary <, <=, =, >=, >, <> belgileriň kömegi bilen ýazylýar.

Her simwola **Set of Char** köplügiň elementi hökmünde seretmek we oňa **in** amalyňy ulanmak bolýar. Mysal üçin, Var

Ch:Charž

...

Ch:='a';

if Ch in ['a'..'z'] then ...

Simwol bahalara we üýtgeýänlere aşakdaky tablisada görkezilen funksiýalar ulanylyp biliner:

Funksiýa : Tipi	Ýerine ýetirýän işi
Chr(X:Byte): Char	ASCII kody X-e deň simwoly berýär
Ord(C:Char): Byte	C simwolyň ASCII kodyny berýär
Pred(C:Char):Char	C –niň önündäki simwoly berýär
Succ(C:Char):Char	C-niň yzyndaky simwoly berýär
UpCase(C:Char):Char	'a'..'z' simwollary degi[lilikde 'A'..'Z' simwollara öwürýär

Bellik.

1.Succ(#255) we Pred(#0) bahalar kesgitlenen dälidir.

2. UpCase funksiýasy diňe setir latyn harplaryny degişli baş latyn harplaryna öwürýär, galan simwollary ökkisi ýaly goýýar.

Setirleri gaýtadan işlemek.

Setirlere baha bermek, birleşdirmek we deňeşdirmek mümkindir. Setirleri birleşdirmek üçin "+" amaly ulanmak bolýar. Bu amala goşmak diýilýär. "+" amaly iki setiri goşýar, Eger goşulanda alnan täze setiriň uzynlygy setiriň bolup biljek uzynlygyndan uly bolsa, onda onuň artyk bölegi zyňylýar.

Setirleri deňeşdirmek setiriň çep tarapyky simwolyndan başlap simwollar boýunça amala aşyrylýar.

Setirleriň üstünde işlemek üçin aşakdaky tablissadaky görkezilen funksiýalar we proseduralar ulanylýar:

Proseduralar we funksiýalar	Ýerine ýetirýän işi
LENGTH(S:String):Byte	S setiriň uzynlygyny tapýar
CONCAT(S1, S2, ..., Sn:String):String	S1,S2, ..., SN -setirleri goşmak
COPY(S:String; Start, Len:Integer):String	S setiriň Start-nji simwolyndan başlap Len sany simwolyňyň copiýasyny alýar
Delete(Var S:String; Start, Len:Integer)	S setiriň Start-nji simwolyndan başlap Len sany simwolyňyň öçürýär
Insert(Var S:String; SubS:String;Start:Integer)	S setiriň Start-nji simwolyndan başlap şol setire SubS setiri goşýar
Pos(SubS,S:string):Byte	S setiriň içinde SubS setiriň bardygyny barlaýar. Eger SubS setir S setiriň içinde bar bolsa, onda onuň S setiriň näçinji simwolyndan başlanýandygyny tapýar, eger SubS setir S setiriň içinde ýok bolsa, onda funksiýa 0 bahany çykarýar.
Str(X[:F[:n]]; Var S:String)	Hakyky ýa-da bitin tipli X sany setir tipli S üýtgeýän ululyga öwürýär. Bu ýerde F - sany ýazmak üçin berilýän öýleriň sany, n -sanyň drob bölegine berilýän öýleriň sany. n parametr diňe X hakyky tipli bolanda ýazylýar
Val(S:String; Var X; Var ErrCode:Integer)	Setir tipli S üýtgeýän ululygynyň bahasyny san görnüşine öwürüp X üýtgeýän ululygynyň öýüne ýazýar. Bu ýerde ErrCode Integer tipli üýtgeýän ululyk. Eger öwürmek üstünlikli geçse ErrCode 0-a deň bolýar Eger öwürmek käbir sebäbe görä amala aşmadyk bolsa, onda ErrCode S setirde öwürmeklige päsgel beren ilkinji simwolyň tertip nomerine deň bolýar. X üýtgeýäniň bahasy bu ýagdaýda üýtgemän galýar.

Bellik. S setiriň başky simwollarynyň boşluk(probel) bolmagy VAL prosedurasý ýerine ýetirilende ýalňyşlyk döretmeýär, ýöne S setiriň soňky simwollarynyň boşluk bolmagy ýalňyşlyk döredýär. Mysal üçin, Val(' 304', n, k) prosedura dogry ýerine ýetiriler. Netijede n-iň bahasy 304, k-nyň bahasy bolsa 0 bolar, Val('304 ' ,n, k) - prosedura ýalňyş bolar. Netijede n-iň bahasy üýtgemez, k-nyň bahasy bolsa 4 deň bolar.

Mysal 1. Setir berlen. Onuň içinde näçe sany probel belgisiniň bardygyny tapmaly.

Çözüşi.

```

Var
S:string;
n, k,i:integer;
begin
    write( 's=');
    read(s);
    n:=length(s);
    k:=0;
for i:=1 to n do
    if s[i]=' ' then k:=k+1;
    write('k=',k);
end.

```

Tekst faýlyndan okamak we ýazmak.

Tekst faýly diýip şu aşakdaky häsiýetlere eýe bolýan faýla aýdylýar:

- a) faýldaky maglumatlar ASCII kodlaryndaky simwollaryň kömegi bilen tekst görnüşinde aňladylýar;
- b) faýldaky maglumatlar setirlere bölünip bilinýär. Setiriň soň bolup #13(kod 13 – CR) simwol hyzmat

edýär. Ol setiri geçiriji #10 (kod 10 – LF) simwol bilen birleşdirilip biliner;

- c) faýlyň soňy ^Z (kod 26) simwol bilen aýdyň bellenýär;
- d) sanlar, setirler we logiki bahalar ýazylanda olar simwol (tekst) görnüşe özgerdilýärler;
- e) sanlar we setirler okalanda olar awtomatiki usulda tekst görnüşinden maşyn görnüşine özgerdilýärler.

Tekst faýllary bilen işlemek üçin TEXT tipli faýl üýtgeýäni kesgitlemeli:

Var

f:Text;

we ony Assign prosedurasynyň kömegi bilen fiziki faýl bilen baglanyşdyrmaly we şondan soň ony açmaly.

Turbo Pascalyň sistema modulynda iki sany: Input we Output atly text-faýl üýtgeýäni (logiki faýl üýtgeýäni) kesgitlenendir. Olar ‘Con’ gurluşy (eger programmada CRT moduly beýan edilen bolsa, onda hyýaly CRT gurluşy) bilen baglanyşdyrylandyrlar. Eger giriş proseduralarynda faýlyň ady ýazylmadyk bolsa, onda giriş Input (bu klawiatura) atly faýlyndan amala aşyrylýar diýilip hasap edilýär. A eger çykyş operatorynda faýlyň ady galdyrylan bolsa, onda çykyş Output faýlyna amala aşyrylýar.

Tekst faýllary bilen işlemek üçin aşadaky funksiýalar we proseduralar ulanylýar:

SetTextBuf(Var f:Text; Var Buf [; BufSize:Word])

prosedurasy f text faýlynyň giriş-çykyş buferini ulaltmak ýa-da kiçeltmek üçin ulanylýar. Buferiň bahasynyň awtomatiki bahasy 128 baýta deňdir. Diskdäki fiziki faýllara ýygy-ýygydan ýüzlenilýän bolsa bu sany birnäçe kilobaýta çenli ulaltmak maslahat berilýär. Beýle etmeklik işi çaltlaşdyrmaga mümkinçilik berýär. Şunlukda gaty diskdäki okaýan/ýazýan gurluş tygşyly ulanylar. Buferi ulaltmaklygy logiki faýl bilen fiziki faýl özara baglanyşdyrylandan soň, ýöne ilkinji okamak ýa-da ýazmak operasiýasyndan öň amala aşyrmaly.

Maglumatlaryň howpsuzlygyny berjaý etmek üçin buferiň ululygyny faýl açylmanka ýerine ýetirmek maslahat berilýär.

SetTextBuf prosedurasynyň kömegi bilen täze bufer döredilende ol prosedura diňe bir f logiki faýly dälde, eýsem Buf üýtgeýäni hem bermelidiris. Şol üýtgeýände bufer ýerleşer. Buf üýtgeýäniň tipi uly rol oýnamaýar. Esasy onuň ölçegi rol oýnaýandyr. Faýlyň buferi Buf üçin berilen ilkinji baýtdan başlanýar we hökmany däl BufSize parametrde görkezilen sana deň bolan baýt ýer eýelär. Eger prosedura çagyrylan wagtynda BufSize san görkezilmedik bolsa, onda ol Buf üýtgeýäniň ölçegine deň diýip hasap edilýär. BufSize-iň bahasy Buf-yň bahasyndan uly görkezilse ol maglumatlaryň ýitirmegine getirer. Ýagny, maglumatlaryň huşdaky Buf üýtgeýän ululygynyň ölçeglerinden çykýan bölegi ýiter.

SetTextBuf prosedurasynyň ulanylyşyna degişli mysallara seredeliň.

Var

ft : Text;

Buf : Array[1..4*1024] of Byte;

Begin

Assign(ft, 'TEXTFILE.DOC'); {logiki faýl fiziki faýl bilen baglanyşdyrylýar}

SetTextBuf (ft, Buf); {bufer üýtgedilýär (indi ol 4 Kb ölçegli Buf) {üýtgeýände}}

Reset(ft);

Read(ft,...);

Reset (ft);

ft faýl bilen başga işler

.....

END.

ft faýl üçin bir gezek kesgitlenen bufer indiki gezek şol ft üýtgeýän üçin SetTextBuf ýa-da Assign proseduralary çagyrylýança öz ýerleşýän ýerini we ölçegini üýtgetmeýär.

Bufer üçin berilýän üýtgeýän ýa global bolmalydyr, ýa-da iň bolmanda berlen faýl bilen iş tamamlanýança saklanmalydyr. Hakykatdanda, eger aşakdaky programmany taýynlasak, onda göräýmäge ähli işler dogry ýerine ýetiriläýjek ýaly bolup görünýär, ýöne ol beýle bolmaz.

```
Prosedure GetFileAndOpenIt ( Var f : Text );
```

```
Var
```

```
Buffer:Array[1..16*1024] of Byte; {Bufer}
```

```
FileName : String;
```

```
Begin
```

```
RedLn(FileName);
```

```
Assign ( f, FileName );
```

```
SetTextBuf (f,Buffer);
```

```
Rewrite(f)
```

```
End;
```

```
Var
```

```
ff : Text;
```

```
Begin
```

```
GetFileAndOpenIt (ff);
```

```
Write(ff, ...); {ff faýla ýazmaga edilýän synanyşyk }
```

```
{Indi näme bolanynyň parhy ýok. Barybir netije nädogry  
bolar}
```

```
end.
```

Bu mysalda açylýan faýlyň buferi Buffer atly üýygeýände ýerleşýär. Ýöne lokal **Buffer** üýtgeýän ululyk diňe prosedura ýerine ýetirilýän wagty bardyr. Prosedura ýerine ýetirilip bolan badyna ol üýtgeýän ululyk ýiter we buferiň ýerleşen oblasty umumy elýeterli bolýar, ol oblastyň çalt wagtda başga bahalar bilen doldurylmagy gaty ähtimaldyr. Şunlukda çagyryan bloga berilýän f (ff) faýlyň özüni nähili alyp barjagy näbellidir. Muňa seretmezden kompilýasiýa edilende-de we sanalýan wagtynda-da hiç hili ýalňyşlyk döremez.

Eger bufer statistiki üýtgeýänede ýerleşdirilse, onda ol maglumatlar oblastynyň ýa-da stegiň bölegini eýelär. A olar bolsa 64 Kb ölçeg bilen çäklenendir, ol bolsa gaty köp däl.

Buferi dinamiki huşda (toplumda) ýerleşdirmek amatly bolardy.

Buferiň ölçegini 512 baýta kratny edip almak maslahat berilýär. Sebäbi bir sektoryň ölçegi 512 baýta deňdir.

Append (Var f :Text) prosedurasy. Bu prosedura tekst faýllaryndaky ýazylan tekste täze tekst goşmak üçin ulanylýar. Bu prosedurany diňe öň bar bolan faýllara ulanyp bolýar. Şunlukda faýl açylýar we onuň soňuna ýazmak ýagdaýynda taýyn bolup durýar. Bu prosedura ulanylanda Rewrite prosedurasyndaky ýaly faýlyň içindäki maglumatlar öçürilmeýär. Eger Append prosedurasyny öň ýok faýla ulanmak üçin synanyşyk edilýän bolsa, onda programmada ýalňyşlyk (hasaba almak ýalňyşlygy) dörär. Tekst faýly Append prosedurasynyň kömegi bilen açylandan soň oňa ýazmaklyk faýlyň soňyny görkezýän belligiň (kod 26) duran yerinden başlanýar. Umuman Append prosedurasy Rewrite prosedurasyndan faýly açmagyň usulyndan (zyyndan) başga hiç zat bilen tapawutlanýan däldir.

Flush (Var f : Text) prosedurasy. Bu prosedura ýazmak üçin (Rewrite we Append proseduralary bilen) açylan faýllara ulanylýar. Maglumatlar ýazylmak üçin ilki buferde toplanýar we ol dolandan soň fiziki faýla ýazylýar. Flush prosedurasy beferiň dolandygyna ýa-da dolmandygyna seretmezden buferdäki maglumatlary fiziki faýla ýazmagy amala aşyrýar. Eger buferiň ölçegi ýeterlik uly bolsa, onda programma sanaýan wagtynda garaşylmadyk ýagdaýda üzülse, buferiň içindäki maglumatlaryň faýla geçmezligi mümkindir. Şunuň öňüni almak üçin programmadaky üzülme bolýjak hasaplanýan yerlerde Flush prosedurasyny çagyrmak gerekdir.

Flush prosedurasy faýly ýapmaýar we çykyşyň zyzgiderligine täsir etmeýär.

EOLn (Var f : Text) : Boolean funksiýasy. Bu funksiýa okamak üçin açylan f tekst faýlyndaky kursoryň duran yerini derňeýär. Funksiýanyň ady End-Of-Line –setiriň soňy diýip terjime edilýär. Eger indiki okaljak simwol setiriň soňyny

görkezýän belgi (#13 simwol) ýa-da faýlyň soňyny görkezýän belgi (#26 simwol) bolsa, onda EOLn funksiýasy TRUE bahany alýar, galan ýagdaýlarda ol FALSE bahany alýar.

EOLn funksiýanyň parametrsiz görnüşi hem bardyr. Bu ýagdaýda ol standart Input operatoryna, başgaça aýdanda klawiaturadan girizmeklige täsir edýär. Bu ýagdaýda EOLn funksiýasy setiriň soňyny görkezýän belginiň öňünden dälde, ol belgiden geçilenden soň (Enter klawiýasy basylan badyna) TRUE bahany alýar. Bu funksiýany ulanyp ENTER klawiýa basylyança pauza goýmak üçin programmada WriteLn(EOLn) operatoryny ýazmak ýeterlikdir.

SeekEOLn (Var f : Text) : Boolean funksiýasy. f tekst faýly diňe okamak üçin açylan bolmalydyr. Eger setiriň soňyna (#13 simwol) ýa-da faýlyň soňyna (#26 simwol) çenli diňe probeller(#32) we(ýa-da) tabulýasiýa (#9 simwol) belgileri galan bolsa, ýa-da indiki okaljak simwol setiriň soňyny görkezýän (#13) ýa-da faýlyň soňyny görkezýän (#26) simwollar bolsa, onda funksiýa TRUE bahany alýar, galan ýagdaýlarda FALSE bahany alýar.

SeekEOF (Var f : Text) : Boolean funksiýasy. f tekst faýly diňe okamak üçin açylan bolmalydyr. Eger indiki okaljak simwol faýlyň soňyny görkezýän (#26) belgi, ýa-da faýlyň soňyna çenli diňe probeller (#32)we (ýa-da) tabulýasiýa (#9) belgileri, we (ýa-da) setiriň soňuny görkezýän (#13) bolsa, onda funksiýa TRUE bahany alýar, galan ýagdaýlarda FALSE bahany alýar.

Turbo Pascalda san we tekst informasiýany girizmek we çykarmak aşakdaky operatorlaryň kömegi bilen amala aşyrylýar:

giriş - Read(f,X) ýa-da Read(f,X1,X2,..., Xn) we
ReadLn(f,X) ýa-da ReadLn(f,X1,X2,..., Xn);
çykyş - Write (f, X) ýa-da Write(f, X1, X2,...,Xn) we
WriteLn (f, X) ýa-da WriteLn(f, X1, X2,...,Xn) .

Eger giriş-çykyş operatorynda birinji parametr bolup faýlyň logiki ady duran bolsa, onda giriş-çykyş şol faýldan bolup

geçýär. Eger giriş-çykyş operatorynda birinji parametr bolup logiki faýl durmaýan bolsa, onda giriş-çykyş Input(klawiaturadan) we Output(Ekrana) standart faýllardan bolup geçer.

Ilki bada giriş operatorlary bolan Read we ReadLn operatorlaryna seredeliň. Bu operatorlaryň argumentleri bolup üýtgeýän ululyklaryň sanawy bolmalydyr. Şol sanawdaky üýtgeýänleriň bahalary faýldan okalýar(ýa-da girizilýär). Tekst faýlyndan okalanda(şol sanda klawiaturadan girizilende hem) üýtgeýän ululyklaryň tipleri diňe bitin, hakyky, simwol(Char), setir (String) we şolar bilen sygyşýan tipler bolmalydyr. Çylşyrymly düzme tipler (massiwler, köplükler, ýazgylar we başgalar) diňe elementler (ýazgylar üçin meýdanlar) boýunça girizilip bilinýändir. Tekst faýlyndan okalanda (şol sanda klawiaturadan okalanda) üýtgeýänleriň bahalary aşakdaky düzgünlere laýyklykda okalýandyr:

- 1) San bahalar girizilende iki sanyň arasynda iň bolmanda bir boşluk(probel), ýa-da tabulýasiýa (#9) simwoly, ýa-da setiriň soňyny görkezýän belgi(#13) bolmalydyr. Klawiaturadan girizilende bahalar ýazylandan soň ENTER klawişasyny basmak hökmandyr.
- 2) Simwol ululyklar girizilende degişli üýtgeýän ululygyň bahasy bolup ön ýanyndaky girizilen simwolyň zyzyndaky simwol alynar. Bu ýerde hiç hili bölüji ýokdur.
- 3) Setiriň bahasynyň okalyşy: şuňa çenli girizilen simwollaryň iň soňkysynyň zyzyndaky simwol setiriň birinji simwoly edilip alynýar. Setir beýan edilende onuň uzynlygy näçä deň bolan bolsa şonça sany simwol okalýar. Ýöne okalýan simwollaryň arasynda #13 simwol gabat gelse, onda setiriň okalmagy gutarýar. Sebäbi #13 simwol setiriň soňyny görkezýär. Bu simwolyň özi setiriň simwoly hökmünde hiç wagt okalmaýar.

Klawiaturadan okalanda Read we ReadLn operatorlarynyň bir-birinden tapawudy ýokdur. Read prosedurasy girizilýän bahany okanda baha okalyp bolandan soň indiki okaljak baha hökmünde şol okalan bahanyň zyndaky bahany belläp goýýar. Indiki gezek Read ýa-da ReadLn prosedurasy çagyrylanda şol bellenen bahadan başlap okalýar. ReadLn (Read Line) prosedurasy bolsa sanawda görkezilen üýtgeýänleriň bahalary okalyp bolandan soň indiki okaljak simwol hökmünde indiki setirdäki birinji simwoly belläp goýýar. Başgaça aýdanda, ReadLn prosedurasy ýerine ýetirilip bolandan soň kursor indiki setire geçýär. ReadLn prosedurasy parametrsiz ýazylsa ol Enter klawişi basylýança pauza bermeklige mümkinçilik berýär.

Tekstiň soňyny görkezýän #26 simwol hem setiri bölüji bolup hyzmat edýär we setiri çäklendirýär. Ýöne ondan soň faýldan okamak mümkin bolmaýar. Sebäbi ol simwol faýlyň soňyny görkezýär. Faýlyň soňyny görkezýän simwol (#26) hiç bir setiriň simwoly hökmünde okalyp bilinmeýär. Ýöne ol simwol Char tipli üýtgeýäniň bahasy hökmünde okalyp bilinýär. #26 simwolyň san baha hökmünde okalmagy 0 sany okanyň bilen deňgüýçlidir.

Write we WriteLn operatorlary X bahany ýa-da X1,X2, ...,Xn bahalaryň sanawyny f tekst faýlyna çykarýarlar. Eger operatorlarda faýl görkezilmedik bolsa, onda bahalar Output faýlyna (displeýiň ekranyna) çykarylýar. Bahalar edil giriş operatoryndaky ýaly diňe bitin, hakyky, simwol we setir tipli ýa-da olaryň önümi görnüşinde bolmalydyrlar. Mümkin bolan düzme tipli üýtgeýänler (ýazgylar, massiwler) meýdanlar ýa-da elementler boýunça çykarylmaladyrlar. Köplükler, görkezjiler(Pointer) , faýllar hem olary ilki bilen çykarylýan komponentlere özgertmezden çykarylyp bilinýän dälidir.

Boolean (Logiki) tipli bahalary çapa çykarmak mümkindir.Ýöne logiki bahalary faýldan edil bolşy ýaly okamak mümkin dälidir. Logiki bahalary faýldan okamak üçin olary san bilen bellemeli:

we False we True bahalaryň deregine 0 we 1 bahalary okamaly. Soňra programmada olary logiki tipli bahalara öwürmeli.

Write prosedurasy sanawda görkezilen bahalary çykarandan soň setiri ýapman goýýar. Indiki gezek Write rosedurasy çagyrylanda ol bahalary öňki setiriň yzyna çykarar (setiri dowam eder). Faýlda setiriň uzynlygy çäklendirilen dälendir. Ýöne tekst displeýiň ekranyna çykarylanda setir ekranýň sag araçägi bilen çäklenendir. Çykarylýan tekst ekranýň sag tarapyna ýetende setir awtomatiki üzülýär we onuň dowamy indiki setire geçýär.

WriteLn prosedurasy sanawda görkezilen bahalary setire çykarýar we soň setiri ýapyp goýýar. Başgaça aýdanda WriteLn prosedurasy setiriň soňuna #13 simwoly goşýar (has dograsy, #13 we #10 simwollary goşýar, ýöne soňky kod esasy kod bilen "birleşýän" ýaly bolýar). Bu bolsa awtomatiki ýagdaýda täze setiri açýar, a ekranda bolsa kursor täze setiriň iň çep tarapky ýerine süýşýär.

Çykarylmalý bahalaryň sanawyny saklamaýan WriteLn ýa-da WriteLn(f) operatory diňe setiriň soňyny görkezýän simwoly saklaýan boş setiri döredýär.

Write we WriteLn operatorlaryň çykyş sanawy hemişelikleri, üýtgeýänleri, aňlatmalary, funksiýalary saklap biler. Olaryň degişli tipleri bolmalydyrlar we otur belgisi bilen aralary açylan bolmalydyr.

Mysal üçin,

```
Write(x, '+', y, '=', x+y);
```

```
Writeln(sin(1.2));
```

Çykyş sanawynda iki sany apostrof "" belgisiniň arasynda ýazylan tekst we setir ýa-da simwol tipli üýtgeýäniň bahasy çapa edil bolşy ýaly edilip çykarylar.

Bitin sanlar hem edil bolşy ýaly çykarylýar. Hakyky sanlar adatça eksponensial görnüşinde çykarylar.

Tekst faýllaryna we ekrana çykarylýan wagtynda bahalary ýazmak üçin ýer(format) görkezmek, başgaça aýdylanda bahalary ýazmak üçin berilýän öýjükleriň (bir öýjük- bir simwolyň ekranda tutýan ýeri) sanyny görkezmek bolýar. Setir we simwol bahalar üçin format bir natural san bilen berilýär we ol san bahanyň yzyndan ikinokat belgisinden soň ýazylýar:

write(a: 2, St:20);

Bu san bahany ýazmak üçin näçe öýjügiň berilýändigini görkezýär. Şeýlelikde, a baha(simwol) bir öýjükde ýerleşip bilýän hem bolsa, oňa 2 öýjük berilýär, St setire bolsa 20 öýjük berilýär. Eger bahanyň tutýan ýeri formatda görkezilen öýjükleriň sanyndan gysga bolsa, onda formatdaky görkezilen artykmaç öýjükler boşluklar bilen doldurylar. Özem boşluklar bahanyň çep tarapynda goýulýar. Ýöne, eger formatda görkezilen san bahany ýerleşdirerden az bolsa, onda baha görkezilen formaty hasaba almazdan çykarylýar. Bu ýagdaýda ýalňyşlyk ýüze çykmaýar. Formatyň ulanylyşynyň şu aýratynlygyny bahalar ekrana çykarylanda gerekli görnüşleri almak üçin ulanmak mümkindir. Käbir ýagdaýda bahany setiriň ortasyndan çykarmak gerek bolýar. Baha ekrana çykarylýan halatynda CRT modulyndaky ýörite prosedurany ulanmak mümkindir, ýöne diskdäki faýla çykarylýan wagtynda ol proseduralar işlemez. A bahany nädip süýşürüp bolar.

Elbetde, onuň üçin

Write(' ',ChapterNameStr);
operatoryny ýazmak bolar. Ýöne ol çözgüt iň gowy çözgüt bolmaýar. Sebäbi programmada şonça sany boşluk(probel) ýöne ýer tutup durar. Dogry çözgüt aşakdaky ýalydyr:
Write(' ':25,ChapterNameStr);

Bu ýerde bir boşluk bar, ýöne ol 25 öýjükde ýazylýar. Netije weli edil öňki ýaly bolar. Soňky ýagdaýda boşlugy aýyryp boş setir hem goýmak bolardy.

FALSE we TRUE bahalar setir hemişelikler hökmünde çykarylýarlar.

Bitin bahalar üçin hem format edil setir bahalardaky ýaly bir natural san bilen berilýär. Mysal üçin,
Write(n:5, 2006:5, (12*9):8);

Bitin san minus alamaty hasaba almak bilen formatda görkezilen öýjüklerde sag tarapdan ýerleşdiriler. Eger öýjükler gereginden artyk bolsa, onda artyk öýjükler çep tarapda galar, olaryň içine boşluklar goýular. A eger-de format gereginden kiçi bolsa, onda ol hasaba alynman baha dogry çykarylar. Format tablissalar çykarylanda amatly bolýar.

Hakyky bahalar üçin format iki sany bitin sanyň kömegi bilen berilýär. Hakyky sanlarda köplenç sanyň drob bölegi hem bolýar. Format görkezilende, adatça bolşy ýaly, ilki bilen bahany ýazmak üçin berilýän öýjükleriň umumy ölçegi ýazylýar, a ondan soň iki nokat goýulyp sanyň drob bölegini ýazmak üçin berilýän öýjükleriň sany ýazylýar:

Write(RealVar:12:3, 2006.01:8:2);

Sanyň hakyky uzynlygy aşakdaky bölekleriň jemine deňdir:

1. alamat üçin bir öýjük berilýär,
2. sanyň bitin bölegini ýazmak üçin gerek öýjükleriň sany,
3. bitin bölegiň yzynda goýulýan nokada bir öýjük berilýär,
4. sanyň drob bölegini ýazmak üçin berilýän öýjükleriň sany.

Mysal üçin, 124322.2323 sany ýazmak üçin format 10:4 görnüşde bolar.

Şeýlelikde, ‘4:3’ görnüşli formatyň manysy ýokdur. Eger format nädogry ýazylan bolsa, onda onuň diňe birinji parametri hasaba alynmaýar, ýöne sanyň oturdan soňky bölegini ýazmak üçin berilýän parametr elmydama dogry hasaba alynýar (tipdäki takyklykdan geçmezlik şerti bilen).

Format görkezilen bolsa sanyň drob bölegi görkezilen ikinji parametre laýyklykda tegeleklenýär. Ýöne üýtgeýän ululygyň öz bahasy üýtgemeýär.

Hakyky sany drob bölegini aýyryp hem çykarmak bolýandyr. Onuň üçin ýazylýan formatda ikinji parametriň bahasy hökmünde 0 goýmak ýeterlikdir.

Mysal üçin, Write(1253.1243:4:0) ýazylsa 1253 çykar.

Hakyky sany eksponensial görnüşde çykarmak üçin formatda diňe bir parametr görkezmek gerekdir. Bu parametr sany ýerleşdirmek üçin berilýän öýleriň uzynlygyny görkezýär. San $-5.5678E+00$ ýa-da $00012E-20$ görnüşi alar. San üçin şeýle format berilende sanyň alamaty üçin öýjük, nokata çenli bir öýjük, nokatdan soň iň bolmanda bir öýjük we dereje üçin dört öýjük – jemi iň azyndan sekiz öýjük berilmelidir.

Eger formatda sekizden kiçi san berilse, onda ol awtomatiki usulda sekize deň diýilip alynýar:
`Write(123.456:8, ' ', 123.456 :6) ;` {şol bir netije alynar}
 Format ulaldylsa sandaky oturdan soňky sifrleriň sany köpeliýär. Olaryň maksimal sany sanyň tipine görä kesgitlenýär, formaty şondan uly edip almak hiç hili peýda bermeýär.

Mysal. $y=x^2$ funksiýanyň $[1,10]$ aralykdaky $dx=1$ ädim bilen tablissasyny işjeň katalogdaky tablissa.txt faýla ýazmaly.

Çözülişi.

```
Var f:text;
x,y,dx:real;
Begin
assign(f,'tablissa.txt');
rewrite(f);
writeln(f,' x    y');
x:=1; dx:=1;
While x<=10 do begin y:=sqr(x); writeln(f, x,y);
End.
```

§16. Simwollaryň ASCII we ANSI kodlary.

Conwertor programmasyny taýynlamak(ASCII \Leftrightarrow ANSI)

ASCII we ANSI kodlary. Kod sahypalary

ASCII(American Standard Code for Information Interchange)- munuň özi maglumatlary alyşmagyň standart amerikan kodydyr. Belli bolşy ýaly kompýuterde ulanylýan

ähli simwollaryň öz kodlary bardyr. Ähli baş we setir latyn harplaryň, sifrleriň we klawiaturadaky beýleki dolandyryjy klawişalaryň kodlary 0-127 aralykda ýerleşýändir. Kod tablissalarynyň(sahypalarynyň) birinji bölegi (128 simwoly) ähli kod tablissalarynda birmeňzeşdir. Kod tablissalarynyň galan böleginde milli elipbiýiň harplary we başga simwollar ýerleşdirilýär. MS DOS operasion sistemasy ASCII kodlaryny ulanýar. Windows operasion sistemasynda bolsa esasan ANSI kodlary ulanylýar. Diske faýl ýazylanda simwollaryň deregine olaryň kodlary ýazylýar. Diskden faýl okalanda bolsa okalan kodlaryň deregine degişli simwollar ekrana çykýar. Biz kähälatlarda MS DOS-da diske ýazylan rus dilindäki teksti Windows operasiýa sistemasyndaky programmalar(mysal üçin Bloknot) bilen okanmyzda düşnüksiz ýazgylary görýäris. Edil şeýle ýagdaý tersine okalanda (Windows-da taýynlanan tekst MS DOS-da okalanda) hem ýüze çykýar. Bu ýagdaý ASCII we ANSI kodlarynyň ikinji bölegikleriniň bir-birine doly laýyk gelmeýändigini sebäpli bolýar.

Conwertor programmasyny taýynlamak(ASCII ⇔ ANSI)

Biz ASCII kodlarynda ýazylan rus tekstini ANSI kodlaryna dogry geçirmek üçin aşakdaky programmany taýynladyk. Onuň üçin ilki bilen MS DOS-da ähli baş we setir rus harplaryny bir tekst faýlyna ýazdyk. Soňra şol faýly Windows-da bloknot programmasy bilen açdyk. Ol ýerde tekstler düşnüksiz tertipde çykýar. Olaryň aşagynda baş we setir rus harplaryny edil MS DOS-daky ýazşymyz ýaly edip ýazýarys we faýly diske ýazýarys. Netijede alnan faýlda iki setir bolýar. Olaryň birinjisini s1 beýlekisini bolsa s2 bilen belleýäris. Indi ASCII kodlarynda ýazylan teksti simwolma-simwol okap başlaýarys. Ol simwollardan s1 setirde gabat gelýänlerini s2 setirdäki degişli simwol bilen çalyşýarys. Netijede aşakdaky programmany alýarys.

{ Rus ASCII to Rus ANSI }

```

uses crt; Label 1;
Var B:Boolean; f1,f2:Text; s1,s2,s3,s4:string; c:char;
begin
s1:='АБВГДЕЖЗИЙКЛМНОПРСТУФХЦШЩЪЫЬЭЮЯ
Ёабвгдежзийклмнопрстуфхцшщъыьэюяё';
s2:='ЂЃ„Ѕ„…†‡€%Љ«ЊЌЋЌЋ»“”•—
□™Љ»Њќћұр ЎўЈѠЈ|§Ё©Є«¬-®Љабвгдежзийклмнопс';
clrscr;
writeln('Enter Name File of ASCII');
readln(s3);
writeln('Enter Name File of ANSI');
readln(s4);
Assign(f1,s3);
Assign(f2,s4);
Reset(s3);
Rewrite(s4);
while not eof(s3) do begin
B:=True;
read(s3,c);
for i:=1 to length(s1) do
if c=s1[i] then begin write(s4,s2[i]); B:=False; end;
if B then write(s4,c); end;
close(s3); close(s4); end.

```

ANSI kodlardan ASCII kodlaryna geçmek

Bu ýagdaýda hem edil öňki algoritme meňzeş algoritmi ulanýarys. Diňe programmadaky s1 we s2 setirleriň ýerlerini çalyşmak ýeterlikdir.

```

{ Rus ANSI to Rus ASCII }
uses crt; Label 1;
Var
B:Boolean; f1,f2:Text;
s1,s2,s3,s4:string; c:char;
begin

```



```

s1:=’АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ
Ёабвгдежзийклмнопрстуфхцчшщъыьэюяё’;
s2:=’ЂЃ„…†‡€%ЉќЊЋЌЇђ‘””•—
□™љњћћр ЎўЈѠЈ§Ё©Є«¬-®Љабвгдежзийклмнопс’;
clrscr;
writeln(‘Enter Name File of ASCII’);
readln(s3);
writeln(‘Enter Name File of ANSI’);
readln(s4);
Assign(f1,s3);
Assign(f2,s4);
Reset(s3);
Rewrite(s4);
while not eof(s3) do begin
B:=True;
read(s3,c);
for i:=1 to length(s1) do
if c=s1[i] then begin write(s4,s2[i]); B:=False; end;
if B then write(s4,c);
end;
close(s3);
close(s4);
end.

```

§17. Maglumatlary goramagyň usullary **Kriptografiýa**

Maglumatlary gorap saklamak üçin dürli serişdeler ulanylýar. Mysal üçin Microsoft Wordda ýazylan dokumentleri parol goýmak bilen goramak mümkindir. Teksti goramagyň ýene-de bir usuly ony şifrlemekdir. Tekst şifrlenende ol ýörite usullar bilen başga tekst bilen çalşyrylýar. Soň ol teksti okamak üçin ony yzyna dikeltmek gerek bolýar. Teksti şeýle usul bilen goramak bilen kriptografiýa ylmy meşgur bolýandyr. Biz

teksti şifrlemegiň we ony yzyna dikeltmegiň usullaryny mysallarda seredip geçeris.

Simwollaryň kodyny bir birlik artdyrmak bilen teksti şifrlemek we ony yzyna dikeltmek

Goý işjeň katalogda **1.txt** tekst faýly berlen bolsun. Ol faýldaky tekstiň her bir simwolynyň kodyny bir birlik ardyryp **2.txt** faýla ýazmaly(şifrlemeli).

Çözülişi.

```
{Shifrleýji programma}
```

```
Var
```

```
f1,f2:Text;
```

```
c:Char;
```

```
k:Integer;
```

```
begin
```

```
Assign(f1,'1.txt');
```

```
Assign(f2,'2.txt');
```

```
Reset(f1);
```

```
Rewrite(f2);
```

```
While not eof(f1) do begin
```

```
read(f1,c);
```

```
k:=ord(c);
```

```
k:=k+1;
```

```
c:=chr(k);
```

```
write(f2,c);
```

```
close(f1);
```

```
close(f2);
```

```
end.
```

Indi bolsa ýokarky programmanyň kömegi bilen şifrlenlen 2.txt faýlyny 1.txt faýlyna dikeldýän programmany ýazalyň. Alnan netijäni 3.txt faýlyna ýazalyň.

```
{Shifrlenlen tekst faýlyny dikeldiji programma}
```

```
Var
```

```

f2,f3:Text;
c:Char;
k:Integer;
begin
Assign(f2,'2.txt');
Assign(f3,'3.txt');
Reset(f2);
Rewrite(f3);
While not eof(f2) do begin
read(f2,c);
k:=ord(c);
k:=k-1;
c:=chr(k);
write(f3,c);
close(f2);
close(f3);
end.

```

Ýörite parołyň kömegi bilen şifrlemek

Aşaky programmada teksti şifrlemekde parol ulanylýar.
 Programma TPI-niň KTU kafedrasynyň uly mugallymy
 O.E.Nurgeldiyew tarapyndan ýazyldy.

USES CRT;

VAR

fi,fo:file of byte;

i,ch:byte; pass:string;

BEGIN

If Paramcount=3 then

begin(*1*)

clrscr;

Assign (fi,paramstr(1)); Reset(fi);

Assign (fo,paramstr(2)); Rewrite(fo);

Pass:=paramstr(3); i:=1;

while not eof (fi) do

```

begin(*2*)
  Read (fi,ch); ch:=ch xor ord(pass[i]);
  inc (i); if i>length(pass) then i:=1;
  write(fo,ch);
end;(*2*)
close(fi); close(fo);
end(*1*)      else write ('Parametr not found'); END.

```

§18. Telefon gözleg ulgamyny taýýarlamak Telefon gözleg ulgamyny döretmek

Telefon gözleg ulgamyna Aşgabat şäheriniň öý telefonlarynyň mysalynda serederis. Öý telefonlary baradaky maglumat şu aşakdaky meýdanlary saklaýar:

1. Telefon nomeri;
2. Müşderiniň familiýasy, A.AA;
3. Poçta indeksi;
4. Köçesi;
5. Jaýy;
6. Korpusy;
7. Öýi.

Maglumatlary saklaýan faýlda meýdanlary biri-birinden "nokatly otur" (;) bilen bölýäris. Faýllaryň adyny SPR_NN.CSV görnüşde ýazýarys, bu ýerde NN – telefon stansiýanyň nomeri. Programma aşakdaky görnüşde bolýar:

```

Uses Crt;
Label 1,2;
Var
number:string[6];
fname,st,fam,indeks,koche, jay,korpus,oy:string;
f:TEXT;

```

```

c:char;
begin
clrscr;
1: writeln('Taze mushderi barada maglumatlary girizin');
write('Telefon nomerini girizin'); readln(number);
write('Müşderiniň familiýasy '); readln(fam);
write('Pochta indeksi '); readln(indeks);
write('Kochesi '); readln(koche);
write('Jay '); readln(jay);
write('Korpus '); readln(korpus);
write('Oy '); readln(oy);
st:=copy(Number,1,2);
fname:='SPR_'+st+'.csv';
Assign(f,fname);
Append(f);
writeln(f,number,',',fam,',',indeks,',',koche,',',jay,
',',korpus,',',oy);
2: writeln('Dowam etmekchimi? Y/N');
read(c);
if c='Y' or c='y' then goto 1 else if c<>'N' or c<>'n' then goto
2;
close(f);
end.

```

Müşderini bazadan aýyrmak

Müşderini bazadan aýyrmak şeýle algoritim boýunça amala aşyrylýar:

1. Müşderiniň telefon nomeri girizilmeli;
2. Degişli faýlyň adyny bellemeli we açmaly. Ondan setirme-setir maglumatlary okamaly;
2. Eger okalyan setirde ýok etmeli nomer ýok bolsa, onda ol setiri täze faýla

- ýazmaly, a eger-de okalan setirde ýok edilmeli nomer bar bolsa, onda ony täze faýla ýazmaly däl;
3. Açylan faýl doly okalyp gutaran bolsa, onda ony ýapmaly we ýok etmeli.
 4. Täze döredilen faýly ýapmaly we onyň adyny öňki ady bellenen faýlyň ady bilen çalyşmaly.

Uses Crt;

Var

number:string[6];

fname:string;

f,f1:TEXT;

begin

clrscr;

writeln('Bazadan aýyrmaly müşderiniň telefon nomerini girizin');

readln(number);

st:=copy(Number,1,2);

fname:='SPR_'+st+'.csv';

fname1:='temp.ttt';

B:=False;

Assign(f,fname);

Reset(f);

Assign(f1,fName1);

rewrite(f1);

while not eof(f) do begin

readln(s);

s1:=copy(s1,1,6);

if s1<>number then writeln(f1,s) else B:=True;

end;

if B then writeln('Abonent bazadan aýryldy') else

writeln('Abonent bazadan tapylmady');

close(f);

close(f1);

Erase(f);

RENAME(f1,fname); end.

Goýlan soraglar boýunça temanyň beýany:

**Telefon nomeri boýunça müşderi baradaky maglumatlary
tapmak**

Meseläniň algoritmi şeýle görnüşde bolýar:

1. Telefon nomeri girizilýär;
2. Programma degişli faýlyň adyny döredýär;
3. Şol faýl işjeň diskden okamak üçin açylýar;
4. Faýldan bir setir okalýar;
5. Okalan setirdäki ilkinji 6 simwoldan düzülen setir girizilen telefon nomeri bilen deňeşdirilýär. Eger olar deň gelse onda okalan setir degişli böleklere bölünip ekrana çykarylýar we programma işini tamamlayar, a eger olar deň gelmeseler 4-nji ädime gaýdyp barylýar we gaýtalamak tä gerekli setir tapylýança, ýa-da faýlyň soňyna ýetilýänçä dowam edýär.

Programma aşakdaky görnüşde bolýar:

Uses Crt;

Label 1,2;

Var

s,st,s1,fname:string;

number,fam,indeks, koche, korpus,jay,oy:String;

f,f1:TEXT;

B:Boolean;

procedure Abonent(s:string; Var

n,fam,ind,koch,j,kor,oy:String);

Var i,k:integer;

a:array[1..7] of string;

begin

k:=1;

for i:=1 to 7 do a[i]:="";

for i:=1 to length(s) do

if s[i]=';' then k:=k+1 else a[k]:=a[k]+s[i];

n:=a[1];

```

fam:=a[2];
ind:=a[3];
koch:=a[4];
kor:=a[5];
j:=a[6];
oy:=a[7];
end;
begin
1: clrscr;
writeln('Telefon nomerini girizin');
readln(number);
if length(number)<>6 then goto 1;
st:=copy(Number,1,2);
fname:='SPR_'+st+'.csv';
B:=False;
Assign(f,fname);
Reset(f);
while not eof(f) do begin
readln(f,s);
s1:=copy(s,1,6);
if s1=number then begin  B:=True;goto 2; end;
end;
2: if B then begin  Abonent(s,number,fam,indeks, koche,
korpus,jay,oy);
writeln('Telefon nomeri ',number);
writeln('Abonentin familiyasy ',fam);
writeln('Pochta indeksi ',indeks);
writeln('Koche ',koche);
writeln('Korpus ',korpus);
writeln('Jay ',jay);
writeln('Oy ',oy);
end else writeln('Telefon tapylmady');
close(f);
end.

```


Müşderiniň familiýasy boýunça telefon nomerini gözlemek

Meseläniň algoritmi şeýle görnüşde bolýar:

1. Müşderiniň familiýasy girizilýär;
2. Girizilen familiýanyň uzynlygy len tapylýar;
3. Programma bazadaky birinji faýly okamak üçin açýar;
4. Faýldan bir setir okalýar;
5. Okalan setiriň 8-nji simwolyndan başlap len sany simwolyndan s1 düzülýär. Ol setir girizilen "müşderiniň familiýasy" bilen deňeşdirilýär. Eger olar deň gelse onda okalan setir degişli bölekler bölünip ekrana çykarylýar, a eger olar deň gelmeseler onda ol setir ekrana çykarylmaýar. Programma ýenede açylan faýldan bir setir okaýar we ýokardaky işi gaýtalaýar. Gaýtalamak tä faýlyň soňyna ýetilýänçä dowam edýär.
6. Eger bazada ýenede faýl bar bolsa onda ol faýl okamak üçin açylýar we 4-nji ädime geçilýär. Eger bazadaky faýllar doly derňelen bolsa onda programma öz işini tamamlýar.

Goý anyklyk üçin telefon bazasynda aşakdaky faýllar bar diýip hasap edeliň:

SPR_31.csv, SPR_32.csv, SPR_33.csv, SPR_34.csv,
SPR_35.csv, SPR_39.csv, SPR_41.csv, SPR_42.csv,
SPR_43.csv, SPR_44.csv, SPR_45.csv, SPR_48.csv,
SPR_49.csv, SPR_50.csv, SPR_51.csv.

Programma aşakdaky görnüşde bolýar:

Uses Crt;

Var

M: set of byte;

s,st,s1,fname:string;

number,fam,indeks, koche, korpus,jay,oy:String;

f,f1:TEXT;

B:Boolean;

len,i:integer;

```

procedure Abonent(s:string; Var
n,fam,ind,koch,j,kor,oy:String);
Var i,k:integer;
a:array[1..7] of string;
begin
k:=1;
for i:=1 to 7 do a[i]:= "";
for i:=1 to length(s) do
if s[i]=';' then k:=k+1 else a[k]:=a[k]+s[i];
n:=a[1]; fam:=a[2]; ind:=a[3]; koch:=a[4]; kor:=a[5];
j:=a[6]; oy:=a[7];
end;
begin
M:=[31,32,33,34,35, 39,41, 42, 43, 44, 45, 48, 49, 50, 51];
clrscr; B:=False;
writeln('Abonentin familiyasyny girizin');
readln(fam); len:=Length(fam);
for i:=1 to 51 do
if i in M then BEGIN
str(i,st);
fname:='SPR_'+st+'.csv';
Assign(f,fname); Reset(f);
while not eof(f) do begin
readln(f,s);
s1:=copy(s,8,len);
if s1=fam then begin B:=True;
Abonent(s,number,fam,indeks, koche, korpus,jay,oy);
writeln('Telefon nomeri ',number);
writeln('Abonentin familiyasy ',fam);
writeln('Pochta indeksi ',indeks);
writeln('Koche ',koche);
writeln('Korpus ',korpus);
writeln('Jay ',jay);
writeln('Oy ',oy);
writeln('-----');

```

```
end; end; close(f); END;  
if not B then writeln('Abonent tapylmady');  
end.
```

§19. Talyplaryň bilimine baha berýän testirleýji programmany taýýarlamak

Okuwçylaryň bilimlerine baha bermek üçin testirlemäni kompýuterde geçirmeklik birnäçe artykmaçlyklara eýedir. Birinjiden ol kagyz çykdaýsyny aradan aýyrýar. Ikinjiden bolsa okuwçy öz bilim derejesini özbaşdak barlamak mümkinçiligini alýar. Üçinjiden baha dogry goýulýar.

Soraglary ýörite tekst faýlynda saklamak amatly bolýar. Faýlyň gurluşy şeýle saýlanýar. Her soraga 20 setir berilýär. Şol 20 setiriň ilkinji 19-sy soragy beýan etmeklik we birnäçe jogap hödürlemek üçin niýetlenýär. 20-nji setire soragda hödürlenen jogaplaryň dogrysynyň nomeri ýazylýar. Anyklyk üçin soraglary saklaýan faýly Test.txt diýip atlandyralyň we ol işjeň katalogda ýerleşen diýip hasap edeliň.

Şeýlelikde Test.txt faýlyndaky setirleriň n sany 20-ä kratny bolýar, soraglaryň sany bolsa $(n \div 20)$ -ä deň bolýar.

Testirleýji programmany düzmek.

Programmanyň algoritmi şeýle bolýar:

1. Soraglary saklaýan faýlyň ady soralýar. Ulanyjy faýlyň adyny girizýär.
2. Programma soraglary saklaýan faýly okamak üçin açýar we onda
näçe setir bardygyny hasaplaýar we ol sany 20-ä bölüp soraglaryň jemi sanyny (ssany) çykarýar;
3. M köplük boşadylýar ($M := []$). Bu ýerde M -okuwça berlen soraglaryň nomerleri).
4. Berlen dogry jogaplaryň sanyny görkezýän ululyk 0 deň edilip alynýar;
5. Soňra 1-ssany aralykdan bir tötän san alynýar.

6. Eger alnan san M köplükde ýok bolsa onda nomeri şol sana deň bolan sorag ekrana

çykarylýar(soraga degişli 20 setiriň diňe 19-sy çykarylýar), soragyň nomeri bolsa

M köplüğe goşulýar.

7. Talyp jogaby girizýär. Ol jogap soraga degişli 20 setiriň 20-nji setirinde ýerleşen

san bilen deňeşdirilýär. Eger olar deň gelseler, onda jogap dogry diýip habar

çykýar we berlen dogry jogaplaryň sany bir birlik artdyrylýar, a eger olar deň

gelmeseler, onda dogry däl diýen habar berilýar;

8. Eger talyba berlen soraglaryň sany 10-dan az bolsa 5-nji ädime dolanyp barylýar.

9. Talybyň beren dogry jogaplaryň sany we oňa berilýän baha ekrana çykarylýar.

Programma aşakdaky görnüşde bolýar:

```
{ Testirleyji programma }
```

```
Uses crt; Label 1,2;
```

```
Var M: Set of byte; fname,ss,s:string; f:TEXT;
```

```
a,ssany,n,k,l,j,l:integer;
```

```
BEGIN clrscr;
```

```
randomize;
```

```
ssany:=0;
```

```
write('Soraglary saklayan faylyn adyny girizin');
```

```
readln(fname);
```

```
assign(f,fname);
```

```
reset(f);
```

```
repeat
```

```
readln(f,s);
```

```
ssany:=ssany+1;
```

```
Until eof(f);
```

```
SSany:=SSany div 20;
```

```
Writeln('Faylda ',SSany,' sorag bar');
```

```
M:=[];
```

```

k:=0; {dogry jogaplaryn sany}
Writeln('Soraglar');
delay(1000);
for i:=1 to 10 do
    BEGIN
    clrscr;
    reset(f);
    writeln(I,'-nji sorag');
    1: n:=random(SSany)+1;
    if n in M then goto 1 else M:=M+[n];
    for j:=1 to (n-1)*20 do readln(f);
    for j:=1 to 19 do begin
        readln(f,s);
        writeln(s);
    end;
    writeln('Dogry jogabyn nomerini yaz! ');
    readln(l);
    readln(f,a);
    if l=a then begin
        writeln('Dogry'); k:=k+1; end else writeln('Dogry dal');
    readln(ss);
    END;
    Writeln('Siz 10 soragdan ',k,' sanysyny bildiniz');
    Case k of
        10 : Writeln('5-lik');
        9 : Writeln('4-lik');
        8 : Writeln('4-lik');
        7 : Writeln('3-lik');
        Else Writeln('2-lik');
    close(f); end.

```

OÝUNLARY PROGRAMMALAŞDYRMAK

§ 20. „X – O“ oýnuny programmalaşdyrmak

Oýunlary programmalaşdyrmakda esasy orun şol oýny oýnamagyň algoritmini düzmeklige degişlidir. Programma oýundaky ýagdaýy seljerip, iň amatly göçümi saýlamalydyr. Häzirki wagtda köp sanly oýunlar üçin dürli programmalar düzüldendir we olar barha kämilleşdirilýär.

Biz “Haç-nol” adyny alan oýny programmalaşdyrmaga synanyşyk etdik. Bu oýunda 3x3 ölçegli boş tablissa berilýär. Oýna iki oýunçy gatnaşýar. Oýunçylar gezekli-gezegine tablissanyň boş öýüne bellik edýärler. Olaryň biri “O”, beýlekisi bolsa “X” belgisini ýazýar. Bir gezekde diňe bir belgi etmek rugsat berilýär. Eger oýunçy belgi edenden soň, tablissanyň sütüninde, ýa-da setirinde, ýa-da diagonalýnda 3 sany meňzeş belgi dursa, onda soňky göçümi eden oýunçy oýuny utýar.

Aşakda şol programma getirilýär:

```
uses crt;  
Label 1,2,3;  
Var  
a:array[1..3,1..3] of char;  
k,n,m,i:integer;  
(*****)  
Function Barla:Boolean;  
Var  
s1,s2,s3,s4,s5,s6,s7,s8,i1,j1:Byte;  
Begin  
Barla:=False;  
For i1:=1 to 3 do begin  
s1:=0; s2:=0; s3:=0;s4:=0;s5:=0;s6:=0;s7:=0;s8:=0;  
for j1:=1 to 3 do begin
```

```

if a[i1,j1]='X' then inc(s1);
if a[j1,i1]='X' then inc(s3);
if a[j1,i1]='0' then inc(s4);
if a[i1,j1]='0' then inc(s2);
if a[j1,j1]='X' then inc(s5);
if a[j1,j1]='0' then inc(s6);
if a[j1,4-j1]='0' then inc(s8);
if a[i1,4-j1]='X' then inc(s7);
end;
if (s7=3)or(s1=3)or(s3=3)or(s5=3) then begin write('Siz
utuldynyz!');
Barla:=True; readkey; halt; end;
if (s8=3)or(s2=3)or(s4=3)or(s6=3) then begin write('Siz
utdunyz!');
Barla:=True; readkey; halt; end;
end;
END;
(*****
Procedure Cap;
Var i1,j1:Byte;
Begin
for i1:=1 to 3 do begin
for j1:=1 to 3 do
write(a[i1,j1], ' ');
writeln;      end;
END;
(*****
Procedure Goy;
Label 4;
Var
max,p,i1,j1:sortint;
s:array[1..8] of shortint;
Begin
for i1:=1 to 3 do begin
for j1:=1 to 8 do s[j1]:=0;

```

```

for j1:=1 to 3 do begin
  if a[j1,i1]='0' then inc(s[4]);
  if a[i1,j1]='0' then inc(s[2]);
  if a[j1,j1]='0' then inc(s[6]);
  if a[j1,4-j1]='0' then inc(s[8]);
  if a[i1,j1]='X' then inc(s[1]);
  if a[j1,i1]='X' then inc(s[3]);
  if a[j1,j1]='X' then inc(s[5]);
  if a[j1,4-j1]='X' then inc(s[7]);
end;
if s[1]=2 then begin for j1:=1 to 3 do
  if a[i1,j1]='' then begin a[i1,j1]:='X'; exit;end;
end;
if s[3]=2 then begin for j1:=1 to 3 do
  if a[j1,i1]='' then begin a[j1,i1]:='X'; exit;end;
end;
if s[5]=2 then begin for j1:=1 to 3 do
  if a[j1,j1]='' then begin a[j1,j1]:='X'; exit;end;
end;
if s[7]=2 then begin for j1:=1 to 3 do
  if a[j1,4-j1]='' then begin a[j1,4-j1]:='X'; exit;end;
end;
if s[2]=2 then begin for j1:=1 to 3 do
  if a[i1,j1]='' then begin a[i1,j1]:='X'; exit;end;
end;
if s[4]=2 then begin for j1:=1 to 3 do
  if a[j1,i1]='' then begin a[j1,i1]:='X'; exit;end;
end;
if s[6]=2 then begin for j1:=1 to 3 do
  if a[j1,j1]='' then begin a[j1,j1]:='X'; exit;end;
end;
if s[8]=2 then begin for j1:=1 to 3 do
  if a[j1,4-j1]='' then begin a[j1,4-j1]:='X'; exit;end;
end;
if (s[1]=0)and (s[3]=0) and(s[5]=0)and(s[7]=0) then

```



```

        if a[2,2]='' then begin a[2,2]:='X'; exit; end;
    end;
    for i1:=1 to 3 do begin
        for j1:=1 to 8 do s[j1]:=0;
        for j1:=1 to 3 do begin
            if a[i1,j1]='X' then inc(s[1]);
            if a[j1,i1]='X' then inc(s[3]);
            if a[j1,j1]='X' then inc(s[5]);
            if a[i1,4-j1]='X' then inc(s[7]);
        end;
        4:
        j1:=3 max:=s[1]; p:=1;
        while j1<=7 do
            begin
                if (max<s[j1])and(s[j1]>=0) then begin max:=s[j1]; p:=j1;
                end;
                j1:=j1+2;
            end;
            if p=1 then begin for j1:=1 to 3 do
                if a[i1,j1]='' then begin a[i1,j1]:='X'; exit; end;
                s[p]:=-1;
            end;
            if p=3 then begin for j1:=1 to 3 do
                if a[j1,i1]='' then begin a[j1,i1]:='X'; exit; end;
                s[p]:=-1;
            end;
            if p=5 then begin for j1:=1 to 3 do
                if a[j1,j1]='' then begin a[j1,j1]:='X'; exit; end;
                s[p]:=-1;
            end;
            if p=7 then begin for j1:=1 to 3 do
                if a[j1,4-j1]='' then begin a[j1,4-j1]:='X'; exit; end;
                s[p]:=-1;
            end;
            goto 4; END; END;

```

```

(*****)
Begin  clrscr;
for i:=1 to 3 do
for k:=1 to 3 do a[i,k]:= '';
write('O'nykim başlaýar? Kompýuter üçin 1, Özüňiz 0
basyň!');
read(k);
if k=0 to thenbegin write('Koordinatalary giriziň!');
read(n,m);
a[n,m]:= '0'; end;
for i:=1 to 4 do begin
goy;
cap;
writeln('-----');
if barla then halt;
2: write('Koordinatalary giriziň!'); read(n,m);
if a[n,m]=' ' then a[n,m]:= '0' else goto 2;
if barla then halt;
cap;
writeln('-----');
end;
if k=1 then begin
for n:=1 to 3 do
for m:=1 to 3 do
if a[n,m]=' ' then a[n,m]:= 'X';
cap;
writeln('-----');
if barla then halt;end;
writeln('Deňlik'); readkey;
END.

```

§21. „NIM“ oýnuny programmalaşdyrmak

Nim oýny iň gadymy we gyzykly matematiki oýunlaryň biridir. Nim oýny aşakdaky düzgünler boýunça oýnalýar:

- Birnäçe hatarda dürli mukdarda daşlar ýerleşýär;
- Oýna iki sany oýunçy gatnaşýar;
- Oýunçylar yzly-yzyna oýnaýarlar. Her oýunçy öz gezegi ýetende haýsy hem bolsa bir hatardan islän möçberinde daş alýar. Bir gezekde birnäçe hatardan daş almak bolmaýar;
- Iň soňky daşy alan oýunçy utýar.

Nim oýny XIX asyryň ahyryna çenli kazinolarda oýnalypdyr. Ýöne XX başynda bu oýun öz meşgurlygyny ýitiripdir. Oňa 1901-nji ýylda Garward uniwersitetiniň matematigi Çarlz L. Butonyň çap eden ylmy makalasy sebäp bolupdyr. Ol nim oýnundaky bolup biljek ýagdaýlary doly derňäp iň amatly strategiýany saýlamagyň usuluny görkezipdir. Bu strategiýany bilýän iki oýunçy nim oýnuny oýnanlarynda daşlaryň başlangyç ýagdaýyna we oýny kimiň başlaýandygyna baglylykda oýnuň netijesi öňünden belli bolupdur. Netijede kazinolarda nim oýnyny oýnamak goýulypdyr. Butonyň tapan iň amatly strategiýasy sanamagyň ikilik sistemasyna esaslanandyr we örän ýönekeýdir. Eger oýunçy nobatdaky göçim edeninden soň daşlaryň ýagdaýy oňa oýunda utmaga doly mümkinçilik berýän bolsa, onda daşlaryň ýagdaýy howpsuz diýilýär, a eger utmaga doly mümkinçilik ýok bolsa, onda ýagdaý howply diýilýär. Islendik howply ýagdaýy bir göçümde howpsuz ýagdaýa getirmek bolýar. Islendik göçüm howpsuz ýagdaýy howply ýagdaýa getirýär. Oýunçy oýunda utmak üçin howply ýagdaýy howpsuz ýagdaýa getirmeli, netijede garşydaşyna daşlaryň ýagdaýyny howply ýagdaýa getirmäge mežbur etmeli. Oýunçy birinji bolup göçýän bolsa we daşlaryň başlangyç ýagdaýy howply bolsa, onda Butonyň strategiýasyny ulanmak oňa utmaga doly mümkinçilik berýär. Eger oýunçy ikinji bolup göçýän bolsa we daşlaryň başlangyç ýagdaýy howpsyz bolsa, onda Butonyň strategiýasyny ulanmak oňa utmaga doly mümkinçilik berýär.

Daşlaryň ýagdaýyny bilmek üçin setirlerdäki daşlaryň sanyny sütünde ýazmaly we ol sanlary ikilik sistema geçirmeli.

Ikilik sistemada ýazylan sanlaryň degişli razrýadlaryny sütünler boýunça goşmaly. Eger sütülerde alnan jemleriň içinde iň bolmanda biri ták san bolsa, onda ol ýagdaý howply bolýar, a ählisi jübüt bolsa, onda ýagdaý howpsuz bolýar.

Butonyň strategiýasyny ulanyp nim oýnuny oýnaýan programma aşakdaky görnüşde bolýar:

Uses CRT

const

MAXROW = 14; {Setirleriň maksimal sany}
MAXCOL = 20; {Setirlerdäki daşlaryň maksimal sany}

type

ColType = array [1..MAXROW] of integer;

var

exit : Boolean; {iş tamamlamagyň nyşany}
change : Boolean; {oýnuň şertiniň üýtgemeginiň nyşany}
nrow : integer; {hatarlaryň sany}
ncol : ColType; {Hatarlar boýunça maksimal daşžaryň mukdary}
col : ColType; {Hatarlar boýunça daşlaryň häzirki ýagdaýy }
{-----}

Procedure ShowField;

const

FISH = #220;
X0 = 4;
X1 = 72;
X = 20;

var

i,j : integer;

begin {ShowField}

for i := 1 to nrow do

begin

GotoXY(X0,i+4);

write(i);

GotoXY(X1,i+4);

```

    write(col[i]:2);
    for j := 1 to ncol[i] do
        begin
            GotoXY(X+2*j,i+4);
            if j<=col[i] then write(FISH) else write('.')
        end
    end
end; {ShowField}
{-----}
Procedure Prepare;
const
    Header0 = 'NIM O'YNY';
    Header1 = 'Siz islendik setirden islendik sany dař alyp
bilersiňiz.';
    Header2 = 'Iň soňky dařy alan oýunçy utýar.';
    Header3 = 'Setiriň nomeri';
    Header4 = 'Dařžaryň sany';
var
    i : integer;
begin {Prepare}
    clrscr;
    GotoXY((80-Length(Header0)) div 2,1);
    write(Header0);
    GotoXY((80-Length(Header1)) div 2,2);
    write(Header1);
    GotoXY((80-Length(Header2)) div 2,3);
    writeln(Header2);
    write(Header3);
    GotoXY(80-Length(Header4),4);
    write(Header4);
    for i := 1 to nrow do col[i] := ncol[i]
end; {Prepare}
{-----}
Procedure GetPlayerMove;
const

```

```

TEXT1 = 'Hatar Mukdary görnüşde öz göçümiňizi giriziň
(mysal üçin, 2 3 – 2-nji hatardan 3 daş almak)';
TEXT2 = 'ýa-da oýundan çykmak üçin 0 0 giriziň; -1 0 oýny
sazlamak üçin';
TEXT3 = 'Siziň göçümiňiz:          ';
Y      = 20;
var
  correctly : Boolean;   x1,x2    : integer;
{-----}
Procedure GetChange;
const
  t1 = 'Oýny sazlamak';
  t2 = '(hatarlaryň sanyny we her hatardaky daşlaryň sanyny
girizmek)';
var
  correctly : Boolean;
  i         : integer;
begin {GetChange}
  clrscr;
  GotoXY((80-Length(t1)) div 2,1);
  write(t1);
  GotoXY((80-Length(t2)) div 2,2);
  write(t2);
  repeat
    GotoXY(1,3);
    write('Setirleriň sanyny giriziň (maksimum ',MAXROW,'):
');
    GotoXY(WhereX-6,WhereY);
    readln(nrow);
    correctly := (nrow<=MAXROW) and (nrow>1);
    if not correctly then
      write(#7)
  until correctly;
  for i := 1 to nrow do
    repeat

```

```

    GotoXY(1,i+3);
    write(' ряд ',i,' Daşlaryň sany
(maksimum ',MAXCOL,'):      ');
    GotoXY(WhereX-6,WhereY);
    readln(ncol[i]);
    correctly := (ncol[i]<=MAXCOL) and (ncol[i]>0);
    if not correctly then
        write(#7)
    until correctly
end; {GetChange}
{-----}
begin {GetPlayerMove}
    ShowField;
    GotoXY((80-Length(TEXT1)) div 2,Y);
    write(TEXT1);
    GotoXY((80-Length(TEXT2)) div 2,Y+1);
    write(TEXT2);
    repeat
        GotoXY(1,Y+2);
        write(TEXT3);
        GotoXY(WhereX-16,Y+2);
        readln(x1,x2);
        exit := x1=0;
        change := x1=-1;
        if not (exit or change) then
            begin
                correctly := (x1>0) and (x1<=nrow) and
                    (x2<=col[x1]) and (x2>0);
                if correctly then
                    begin
                        col[x1] := col[x1]-x2;          ShowField
                    end
                else
                    write(#7)
            end
        end
    end
end

```

```

    else
        correctly := true
    until correctly;
    if change then
        GetChange
end; {GetPlayerMove}
{-----}
Procedure SetOwnerMove;
{-----}
FUNCTION CheckField : integer;
var
    i,j : integer;
begin {CheckField}
    j := 0;
    for i := 1 to nrow do if col[i]>0 then inc(j);
        CheckField := j
    end; {CheckField}
{-----}
Procedure CheckPlay;
var
    i : integer;
begin {CheckPlay}
    GotoXY(1,25);
    write('Ýenede oýnamak üçin 1,çýkmak üçin 0 giriziň:');
    readln(i);
    if i=1 then change := true else exit := true
end; {CheckPlay}
{-----}
Procedure PlayerVictory;
const
    t1 = 'Gowy oýun bilen gutlaýaryn!';
var i : integer;
begin
    GotoXY((80-Length(t1)) div 2,24);
    writeln(t1,#7);

```



```

for i := 1 to nrow do
  if ncol[i]<MAXROW then inc(ncol[i]);
  CheckPlay
end; {PlayerVictory}
{-----}
Procedure OwnVictory;
const
  t1 = 'Siz utuldyňyz: indiki göçümde men tutuş hatary alýan ';
var
  i : integer;
begin {OwnVictory}
  i := 1;
  while col[i]=0 do inc(i);
  GotoXY((80-Length(t1)) div 2,24);
  write(t1,i,#7);
  delay(2000);
  col[i] := 0;
  ShowField;
  CheckPlay
end; {OwnVictory}
{-----}
Procedure ChooseMove;
const
  BIT = 6;
type
  BitType = array [1..BIT] of integer;
var
  ncbit : array [1..MAXROW] of BitType;
  i,j,k : integer;
  nbit : BitType;
{-----}
Procedure BitForm(n : integer; var b : BitType);
var
  i : integer;
begin {BitForm}

```

```

for i := BIT downto 1 do
  begin
    if odd(n) then b[i] := 1 else b[i] := 0;
    n := n shr 1
  end
end; {BitForm}
{-----}
begin {ChooseMove}
  for i := 1 to nrow do BitForm(col[i],ncbit[i]);
  for i := 1 to BIT do
    begin
      nbit[i] := 0;
      for j := 1 to nrow do nbit[i] := nbit[i] xor ncbit[j,i]
    end;
    i := 1;
    while nbit[i]=0 do inc(i);
    if i>BIT then
      begin
        j := 1;
        while col[j]=0 do inc(j);
        k := 1
      end
    else
      begin
        j := 1;
        while ncbit[j,i]=0 do inc(j);
        for i := i to BIT do
          if nbit[i]=1 then
            ncbit[j,i] := ord(ncbit[j,i]=0);
          k := 0;
          for i := 1 to BIT do
            begin
              if ncbit[j,i]=1 then inc(k);
              if i<BIT then k := k shl 1
            end;

```

```

    k := col[j] - k
end;
GotoXY(1,23);
write('Menin göçümim:      ');
GotoXY(WhereX-8,WhereY);
delay(1000);
write(j,',k);
col[j] := col[j]-k
end; {ChooseMove}
{-----}
begin {SetOwnerMove}
  case CheckField of
    0 : PlayerVictory;
    1 : OwnVictory;
  else
    ChooseMove;
  end; {case}
end; {SetOwnerMove}
{-----}
begin
  nrow := 3;
  ncol[1] := 3;
  ncol[2] := 4;
  ncol[3] := 5;
  repeat
    Prepare;
    repeat
      GetPlayerMove;
      if not (exit or change) then
        SetOwnerMove
      until exit or change
    until exit
  end.

```

§22. Küşt oýny bilen baglanyşykly meseleleri programmalaşdyrmak

Mesele 1. 8X8 ölçegli küşt tagtasynda her bir öýjüğe diňe bir gezek baryp atyň göçümi bilen aýlanyp çykmany. Başlangyç öýjügiň koordinatalary berilýär. Her göçümde öjüğe artýan tertipde 1,..., 64 sanlaryň degişlisini ýazmaly.

Çözülişi.

Bu meseläni birnäçe usul bilen işlemek bolar. Ilki bilen bir şözüwi özümiz elimiz bilen tapýarys we şol ýoly ýatda saklaýarys. Koordinatalary berilen nokatda 1 ýazýarys we öň ýatda saklanan ýol boýunça öýjüklere sanlary ýazmagymyzy dowam edýäris.

Programma aşakdaky görnüşde bolýar.

```
uses crt; { Bayram A. }  
label 1;  
Var  
a:array[1..8,1..8] of byte;  
x,y:array[1..64] of byte;  
t,x0,y0,k,i,j:integer;  
begin  
clrscr;  
x[1]:=1; x[2]:=3; x[3]:=5; x[4]:=7; x[5]:=8;  
x[6]:=7; x[7]:=8; x[8]:=6; x[9]:=4; x[10]:=2;  
x[11]:=1; x[12]:=3;x[13]:=1;x[14]:=3;  
x[15]:=5; x[16]:=7;x[17]:=8;x[18]:=7;x[19]:=8;  
x[20]:=6;x[21]:=4;x[22]:=2;x[23]:=1;x[24]:=2;  
x[25]:=3;x[26]:=1;x[27]:=2;x[28]:=1;x[29]:=2;  
x[30]:=4;x[31]:=6;x[32]:=8;x[33]:=7;x[34]:=8;  
x[35]:=7;x[36]:=5;x[37]:=6;x[38]:=8;x[39]:=6;  
x[40]:=4;x[41]:=2;x[42]:=1;x[43]:=2;x[44]:=1;  
x[45]:=3;x[46]:=5;x[47]:=7;x[48]:=8;x[49]:=6;  
x[50]:=5;x[51]:=7;x[52]:=5;x[53]:=3;x[54]:=4;  
x[55]:=5;x[56]:=3;x[57]:=4;x[58]:=6;x[59]:=4;  
x[60]:=6;x[61]:=5;x[62]:=4;x[63]:=3;x[64]:=2;
```

```

y[1]:=1;y[22]:=1;y[45]:=1;y[30]:=1;
y[3]:=1;y[20]:=1;y[47]:=1;y[32]:=1;
y[44]:=2;y[29]:=2;y[2]:=2;y[21]:=2;
y[46]:=2;y[31]:=2;y[4]:=2;y[19]:=2;
y[23]:=3;y[64]:=3;y[53]:=3;y[62]:=3;
y[55]:=3;y[60]:=3;y[33]:=3;y[48]:=3;
y[28]:=4;y[43]:=4;y[56]:=4;y[59]:=4;
y[52]:=4;y[49]:=4;y[18]:=4;y[5]:=4;
y[11]:=5;y[24]:=5;y[63]:=5;y[54]:=5;
y[61]:=5;y[58]:=5;y[51]:=5;y[34]:=5;
y[42]:=6;y[27]:=6;y[12]:=6;y[57]:=6;
y[50]:=6;y[37]:=6;y[6]:=6;y[17]:=6;
y[13]:=7;y[10]:=7;y[25]:=7;y[40]:=7;
y[15]:=7;y[8]:=7;y[35]:=7;y[38]:=7;
y[26]:=8;y[41]:=8;y[14]:=8;y[9]:=8;
y[36]:=8;y[39]:=8;y[16]:=8;y[7]:=8;
writeln('x,y=');readln(x0,y0);
k:=0;
for i:=1 to 64 do
if (x[i]=x0) and(y[i]=y0) then begin t:=i; goto 1;end;
1: for i:=t to 64 do begin k:=k+1; a[x[i],y[i]]:=k; end;
  for i:=1 to t-1 do begin k:=k+1; a[x[i],y[i]]:=k; end;
WRITELN('Atyn gochumleri');
for i:=1 to 8 do begin writeln;
for j:=1 to 8 do write(a[i,j]:4, ' ');
  end;
readln;
end.

```

Şu meseläniň çözülişiniň ýene bir görnüşini getireliň.

Aşakdaky programmada atyň göçümleri ön taýyn ýol boýunça tapylan, eýsem derňew netijesinde tapylýar. Birinji ädimde 1 dälde 0 goýulýar.

```

uses crt;
type m = array[1..8,1..8] of integer;
const a : m=((0, 43, 22, 27, 10, 41, 12, 25),

```

```

(21, 28, 63, 42, 23, 26, 9, 40),
(44, 1, 52, 55, 62, 11, 24, 13),
(29, 20, 61, 58, 53, 56, 39, 8),
(2, 45, 54, 51, 60, 49, 14, 35),
(19, 30, 59, 48, 57, 36, 7, 38),
(46, 3, 32, 17, 50, 5, 34, 15),
(31, 18, 47, 4, 33, 16, 37, 6));

var
  b : m;
i,j,x,y,c : integer;
BEGIN
  clrscr;
  write('x='); read(x);
  write('y='); read(y);
  for i:=1 to 8 do
    begin(*1*)
      for j:=1 to 8 do
        begin(*2*)
          c:=a[i,j]-a[x,y];
          if(c<0) then c:=c+64;
          b[i,j]:=c;
          write(b[i,j]:4);
        end;(*2*)
      writeln;
    end;(*1*)
  readln; readln;
END.

```

GRAFIKI EKRANY DOLANDYRMAK

§ 23. Displeý adapterleriniň grafiki iş kadalarynyň gysgaça häsiýetnamasy

Graph moduly ekrana şekilleri çykarmak üçin ulanylýan serişdeleri saklaýar. Modulda jemi 50-den gowrak funsiýalar we proseduralar bardyr. Turbo Paskal işe göýberilende ekran tekst kadasynda açylýar. Şonuň üçin kompýuteriň grafiki mümkinçiliklerini ulanýan islendik programma displeý adapteriniň grafiki iş kadasyny gurnamalydyr.

Grafiki proseduralary anyk adapter bilen işlemek üçin sazlamak daşky draýwerleriň kömegi bilen amala aşyrylýar. Draýwer – munuň özi personal kompýuteriň tehniki serişdesini dolandyryýan programmadyr. Grafiki draýwer displeý adapterini grafiki kadada dolandyryar. Grafiki draýwerler diskde *BGI* (*Borland Graphics Interface* – inlis sözünden alynýar) katalogda BGI giňeltmeli faýllar görnüşinde saklanýarlar. Mysal üçin, CGA.BGI - CGA adapter üçin draýwer, EGA.VGA.BGI – EGA we VGA adapterler üçin draýwer we ş.m.

Häzirki wagtda öndürilýän personal kompýuterler IBM firmasy tarapyndan işlenip taýynlanan adapterler ýa-da şolar bilen kybapdaş adapterler bilen enjamlaşdyrylýar. Öňki sapaklarda ýatlanyp geçilen monohrom MDA adapterini hasaba almasak, olaryň ählisi grafiki kadada işläp bilýändirler. Bu kadada displeýiň ekraný bir-birine örän golaý ýerleşen nokatlaryň – pikselleriň toplumy hökmünde seredilýär. Ol nokatlaryň her biriniň belli bir reňkli şöhlelenmegini programma tarapyndan dolandyrmak bolýar.

Anyk adapteriň grafiki mümkinçilikleri ekranýň mümkinçilikleri bilen kesgitlenýär, başgaça aýdanda pikselleriň umumy sany, şeýle hem olaryň islendiginiň

şöhledenip biljek reňkleriniň umumy sany bilen kesgitlenýär. Ondan başgada köp adapterleriň birnäçe grafiki sahypa bilen işlemek mümkinçilikleri bardyr. Grafiki sahypa diýip operatiw huşuň ekranyň "kartasyny" döretmek üçin ulanylýan oblastyna (başgaça aýdanda operatiw huşuň ekranyň her bir pikseliň reňki baradaky maglumaty saklaýan oblastyna) aýdylýar. Aşakda has köp ýaýran adapterleriň grafiki iş kadalarynyň gysgaça häsiýetnamalary getirilýär.

CGA (Color Graphics Adapter – reňkli grafiki adapter) adapteriň 5 sany grafiki kadasy bardyr. Olardan 4 kada ekranyň pes ygtyýarlyk mümkinçiligine laýyk gelýär (keseligine 320 piksel, dikligine 200 piksel, b.a. 320x200) we diňe ulanylýan reňkleriň toplumy – palitrasy bilen tapawutlanýarlar. Her bir palitra 3 reňkden durýar, a şöhlenenmeýän pikselleriň gara reňki hem hasaba alynanda bolsa 4 reňkden ybaratdyr: palitra 0 (ýagty ýaşyl, gülgüne, sary), palitra 1 (ýagty mawy, gyrgyz, ak), palitra 2 (ýaşyl, gyzy, mele) we palitra 3 (mawy, syýareňk, ýagty-çal). 5-nji kada ýokary 640x200 ygtyýarlyga laýyk gelýär, ýöne bu ýagdaýda her bir piksel öň saýlanan haýsy hem bolsa bir we ähli pikseller üçin meňzeş bolan reňk bilen şöhledenip, ýa-da hiç hili şöhlenenmän biler. Başgaça aýdanda, 5-nji kadanyň palitrasynda diňe iki sany reňk bardyr. Grafiki kadada CGA adapteri diňe bir sahypany ulanýar.

EGA (Enhanced Graphics Adapter – güýçlendirilen grafiki adapter) adapteri CGA adapteriň grafiki kadalaryny doly goldaýar. Ondan başgada pes ygtyýarlykly (640x200, 16 reňk, 4 sahypa) we ýokary ygtyýarlykly (640x350, 16 reňk, 1 sahypa) kadalary bardyr. EGA adapteriniň käbir görnüşlerinde monohrom kadasy (640x350, 1 sahypa, 2 reňk) hem ulanylýandyr.

MCGA (Multi-Color Graphics Adapter – köpreňkli grafiki adapter) CGA adapter bilen sygyşýandyr we ýenede bir kada (640x480, 2 reňk, 1 sahypa) eýedir.

VGA (Video Graphics Adapter – grafiki video adapter) adapteri CGA we EGA adapterleriň kadalaryny goldaýandyr

we ondan başgada ýokary ygtyýarlykly kada (640x480, 16 reňk, 1 sahypa) eýedir.

SVGA (Super Video Graphics Adapter –super grafiki video adapter) adapteri 800x600 we ondan hem ýokary ygtyýarlygy, 256 we ondan hem köp reňki ulanýan adapterdir. Ýöne Turbo Pascal 7.0 –da bu adapteriň draýweri ýokdur. Onuň üçin EGA VGA.BGI draýwerini ulanmaly bolýar.

§24. Grafiki kadada işlemek üçin ulanylýan proseduralar we funksiýalar. Figuralary hereketlendirmek

InitGraph prosedurasy. Adapteriň grafiki iş kadasyny gurnaýar.

Proseduranyň sözbaşysy:

Procedure InitGraph (var Driver, Mode: Integer; Path : String);

Bu ýerde *Driver* – grafiki draýweriň tipini kesgitleýän *Integer* tipli üýtgeýän ululyk; *Mode* - grafiki adapteriň iş kadasyny kesgitleýär; *Path* - draýweriň faýlynyň adyny(kähalatlarda faýlyň doly marşrutyny) saklaýan setir tipli üýtgeýän ululykdyr.

Prosedura çagyrylmazyndan öň kompýuterdäki diskleriň birinde gerekli grafiki draýweri saklaýan faýl bolmalydyr. *InitGraph* prosedurasy bu draýweri operatiw huşa ýükleyär we adapteri grafiki iş kadasyna geçirýär. Draýweriň tipi grafiki adapteriň tipine laýyk gelmelidir. Draýweriň tipini kesgitlemek üçin Graph modulynda aşakdaky hemişelikler kesgittenendir:

const

Detect = 0; {Awtokesgitlemek tipiniň kadasy}

CGA =1;

MCGA =2;

EGA =3;

EGA64 =4;

EGAMono =5;

IBM8514 =6;

```

HercMono    =7;
ATT400      =8;
VGA         =9;
PC3270      =10;

```

Adapterlerin köpüsü dürli kadalarda işläp bilýärler. Adaptere gerekli kadany görkezmek üçin *Mode* üýtgeýäni ulanylýar. *Mode* üýtgeýäniň bahasy hökmünde aşakdaky hemişelikleri ulanmak mümkindir:

const

{ Adapter CGA: }

```

CGACO       = 0; {Pes ygtyýarlyk, palitra 0}
CGAC1       = 1; {Pes ygtyýarlyk, palitra 1}
CGAC2       = 2; {Pes ygtyýarlyk, palitra 2}
CGAC3       = 3; {Pes ygtyýarlyk, palitra 3}
CGAHi       = 4; {Ýokary ygtyýarlyk}

```

{ Adapter MCGA: }

```

MCGAC0      = 0; {CGAC0 – kadasy ýaly}
MCGAC1      = 1; {CGAC1 – kadasy ýaly}
MCGAC2      = 2; {CGAC2 – kadasy ýaly}
MCGAC3      = 3; {CGAC3 – kadasy ýaly}
MCGAMed     = 4; {CGAHi – kadasy ýaly}
MCGAHi      = 5; {640x480 }

```

{ Adapter EGA }

```

EGALo       = 0; {640x200, 16 reňk}
EGAHi       = 1; {640x350, 16 reňk}
EGAMonoHi   = 3; {640x350, 2 reňk}

```

{ HGC we HGC+ adapterler: }

```

HercMonoHi  == 0; {720x348}

```

{ ATT400 adapteri: }

```

ATT400C0    =0; {CGAC0 kadanyň analogy}
ATT400C0    =1; {CGAC1 kadanyň analogy}
ATT400C0    =2; {CGAC2 kadanyň analogy}
ATT400C0    =3; {CGAC3 kadanyň analogy}
ATT400C0    =4; {CGAC4 kadanyň analogy}

```

```

ATT400C0                                =5; {640x400, 2 reňk}
{VGA adapteri:}
VGALo=0;                                {640x200}
VGAMed=1;                               {640x350}
VGAHi=2;                                {640x480}
PC3270Hi=0;                             {HercMonoHi –yň analogy}
{IBM8514 adapteri:}
IBM8514Lo=0;                             {640x480, 256 reňk}
IBM8514Hi=1;                             {1024x768, 256 reňk}
Goý CGA.BGI draýwer C:\TP\BGI katalogda ýerleşýän bolsun
we 2 reňkli 320x200 iş kadasy gurnalýan bolsun. Onda
InitGraph prosedurasyna ýüzlenmeklik aşakdaky ýaly bolar:
Uses Graph;
Var
Driver, Mode:Integer;
Begin
Driver:=CGA;
Mode:=CGAC2;
InitGraph(Driver, Mode, 'C:\TP\BGI');

```

....

Eger personal kompýuteriň adapteriniň tipi näbelli bolsa ýa-da programma islendik adapter bilen işlemägä niýetlenen bolsa, onda InitGraph prosedurasyna ýüzlenmeklik aşakdaky ýaly bolar:

```
Driver:=Detect;
```

```
InitGraph(Driver, Mode, 'C:\TP\BGI');
```

SetColor prosedurasy. Bu prosedura çykarylýan çyzyklaryň we simwollaryň reňkini kesgitleýär. Sözbaşysy:

Procedure SetColor(Color:Word);

Bu ýerde Color – reňkiň nomeri ýa-da ady.

Graph modulynda reňki görkezmek üçin edil CRT modulyndaky ýaly hemişelikler kesgitlenendir.

CloseGraph prosedurasy. Bu prosedura adapteriň grafiki iş kadasyňy ýatyrýar we ekranyň tekst iş kadasyňy dikeldýär.

Prosedura parametrsiz çagyrylýar: **CloseGraph;**

Adapteriň grafiki iş kadasynda ekran piksellerden durýar. Ekrandaky nokadyň kordinatalary ýokarky çep burça görä alynýar. Ekranýň iň ýokarky çep burçynyň koordinatalary (0,0) diýip hasaplanýar. Nokadyň birinji koordinatasy onuň kese koordinatasyny, a ikinji koordinatasy bolsa, dik koordinatasyny aňladýar.

GetMaxX we GetMaxY funksiýalary. Bu funksiýalar Word tipli bahany alýarlar we degişlilikde ekranýň maksimal kese we maksimal dik koordinatasyny görkezýarlar.

GetX we GetY funksiýalary. Bu funksiýalar Integer tipli bahany alýarlar we degişlilikde görkezijiniň duran ýeriniň kese we dik koordinatasyny görkezýarlar.

MoveTo prosedurasy. Görkezijini görkezilen nokada geçirýär.

Proseduranyň sözbaşysy:

Procedure MoveTo(X,Y:Integer);

Bu ýerde X,Y – görkezijiniň geçmeli ýeriniň koordinatalary.

MoveRel prosedurasy. Görkezijini öňki duran ýerine görä täze nokada geçirýär.

Proseduranyň sözbaşysy:

Procedure MoveRel(DX,DY:Integer);

Bu ýerde DX,DY –koordinatalara berilýän artdyrma.

ClearDevice prosedurasy. Grafiki ekraný arassalaýar.

Prosedura ýüzlenilenden soň görkeziji (0,0) nokada geçýär, ekran bolsa fonyň reňki bilen reňklenýär. Fonuň reňki bolsa SetBkColor prosedurasynyň kömegi bilen kesgitlenýär.

Proseduranyň sözbaşysy:

Procedure ClearDevice;

Çyzyklar we nokatlar. Köpburçluklar. Dugalar, töwerekler we ellipsler.

PutPixel prosedurasy. Bu prosedura berlen koordinatalar boýunça berlen reňkli nokady ekrana çykarýar. Proseduranyň sözbaşysy:

Procedure PutPixel(X,Y:Integer; Color :Word);

GetPixel funksiyasy. Bu funksiya berlen koordinatalarda ýerleşýän nokadyň reňkini berýär. Funksiýanyň sözbaşysy:

Function GetPixel(X,Y:Integer) :Word;

Line prosedurasy. Bu prosedura başynyň we soňunyň berlen koordinatalary boýunça çyzygy ekrana çykarýar. Proseduranyň sözbaşysy:

Procedure Line(X1,Y1, X2,Y2:Integer);

LineTo prosedurasy. Bu prosedura görkezijiniň öňki duran ýerinden başlanýan we koordinatalary berlen nokatda gutarýan çyzygy ekrana çykarýar. Proseduranyň sözbaşysy:

Procedure LineTo(X,Y:Integer);

LineRel prosedurasy. Bu prosedura görkezijiniň öňki duran ýerinden başlanýan we gutarýan ýeriniň koordinatalary berlen artdyrmalara görä kesgitlenýän çyzygy ekrana çykarýar.

Proseduranyň sözbaşysy:

Procedure LineRel(DX,DY:Integer);

Rectangle prosedurasy. Bu prosedura depeleriniň berlen koordinatalary boýunça gönüburçlyk gurýar.

Proseduranyň sözbaşysy:

Procedure Rectangle(X1,Y1,X2,Y2: Integer);

Bu ýerde (X1,Y1) gönüburçlugyň ýokarky çep depesiniň koordinatalary, (X2,Y2) gönüburçlugyň aşaky sag depesiniň koordinatalary.

Circle prosedurasy. Bu prosedura töwerek gurýar.

Proseduranyň sözbaşysy:

Procedure Circle(X,Y: Integer;R:Word);

Bu ýerde (X,Y) töweregiň merkeziniň koordinatalary, R – töweregiň radiusy.

Arc prosedurasy. Bu prosedura töweregiň dugasyny gurýar.

Proseduranyň sözbaşysy:

Procedure Arc(X,Y: Integer;BegA,EndA,R:Word);

Bu ýerde (X,Y) merkeziniň koordinatalary, BegA,EndA - duganyň başlangyç we gutarýan burçy,R – radius.

Burçlar sagat strelkasynyň tersine sanalýar we graduslarda görkezilýär. Nul burç çepden saga ugrukdyrylan kese wektoryň ugruna laýyk gelýär.

Ellipse prosedurasy. Bu prosedura ellipsiň dugasyny gurýar. Proseduranyň sözbaşysy:

Procedure Ellipse(X,Y: Integer;BegA,EndA,RX,RY:Word);

Bu ýerde (X,Y) merkeziniň koordinatalary, BegA,EndA - duganyň başlangyç we gutarýan burçy,RX,RY – degişlilikde ellipsiň kese we dik radyuslarynyň pikseldeki bahalary.

Aşakdaky pogramma ekranyň diagonallaryny çyzar.

Uses Graph;

Var

gd,gm:Integer;

begin

gd:=detect;

InitGraph(gd,gm,'D:\TP\BGI');

Line(0,0,GetmaxX,GetMaxy);

Line(0,GetMaxy,GetmaxX,0);

readln; closeGraph;

end.

Ektranda figuralary hereketlendirmek aşakdaky usul bilen amala aşyrylýar:

1. Figura çyzylýar;
2. Ol figura Delay prosedurasynyň kömegi bilen az wagtranda saklanýar;
3. Figura fonyň reňkinde täzeden çyzylýar. Netijede ol ekrandan ýitýär.
4. Figura öňki duran ýerine görä azrak süýşürilip täzeden fonyň reňkinden başga reňkde ekranda çyzylýar;
5. 2-nji punkta geňilýär.

Figuralary hereketlendirmäge degişli aşakdaky meselä seredeliň.

Mesele 1. Dik çyzygy ekranyň çep tarapyndan sag tarapyna hereketlendirmeli.

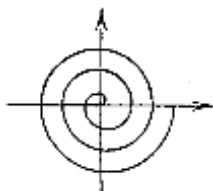
Meseläniň TurboPaskal dilindäki programmasy aşakdaky görnüşde bolar:

```
Uses crt,graph;
Var
n,k,gd,gm:integer;
begin
gd:=0;
  initgraph(gd,gm,"");
  for n:=1 to 500 do begin
    setcolor(white);
    line(n,0,n,200);
    delay(1000);
    setcolor(Black);
    line(n,0,n,200);

                                end;
  closegraph;

end.
```

Mesele 2. Ekranda Arhimiדין spiralynyň şekilini gurmaly.



Programmasy:

```
uses crt,graph;
var
  gd,gm,n,alfa,x,y : integer;
  t,r : real;
begin
  write('n='); read(n);
  write('r='); read(r);
  write('Alfa='); read(alfa);
```

```

t:=alfa;

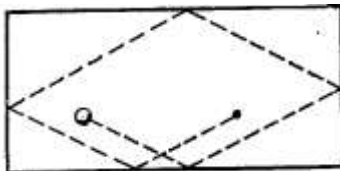
gd:=detect; Initgraph(gd,gm,'e:\lang\tp7_1\bgi');
moveTo(320,240);
repeat
    t:=t+0.0001;
    r:=5*t;
    x:=320-round(r*cos(t)); y:=240+round(r*sin(t));
    lineTo(x,y);

until t>2*n*pi;

readkey;
closegraph;
end.

```

Mesele 3. Ekranda dynuwsyz hereket edýän billiard şarynyň şeklini gurmaly.



Programmasy:

```

uses crt,graph;
var d,r,x,y : integer;
    lx,ly : boolean;

begin
d:=0; initgraph(d,r,'d:\olimp\tp7_1\bgi');
randomize;
    x:=50; y:=0;
    repeat
        setcolor(15);

```



```

        circle(x,y,10);
        delay(2000);
        setcolor(0);
        circle(x,y,10);
        if (x=0) or (x=639) then lx:=not lx;
        if (y=0) or (y=479) then ly:=not ly;
        if lx then inc(x) else dec(x);
        if ly then inc(y) else dec(y);
    until keypressed;
readkey;
closegraph;
end.

```

§25. Grafiki ekrana tekst çykarmak.

Grafiki ekrana tekst çykarmak üçin aşakdaky proseduralar ulanylýar.

OutText prosedurasy. Bu prosedura görkezijiniň duran ýerinden başlap tekst setirini çykarýar.

Sözbaşysy:

Procedure OutText (Txt:String);

Bu ýerde Txt – çap edilýän setir.

OutTextXY prosedurasy. Bu prosedura görkezilen ýerden başlap tekst setirini çykarýar.

Sözbaşysy:

Procedure OutTextXY (X,Y:Integer; Txt:String);

Bu ýerde X,Y – çap edilmeli ýeriň koordinatalary, Txt – çap edilýän setir.

SetTextStyle prosedurasy. Bu prosedura grafiki ekrana tekst çykarmagyň stilini gurnaýar.

Sözbaşysy:

Procedure SetTextStyle (Font, Direct, Size:Word);

Bu ýerde Font – şriftiň kody(nomeri); Direct – ýazylyş ugurynyň kody, Size – şriftiň ölçeginiň kody.

Şriftiň kodyny görkezmek üçin Graph modulynda kesgitlenen aşakdaky hemişelikleri ulanmak bolýar:

Const

DefaultFont	=0; {Nokatlanç şrift }
TriplexFont	=1; {Üçeldilen TRIP.CHR şrifti }
SmallFont	=2; {Kiçeldilen LITT.CHR şrifti }
SansSerifFont	=3; {Göni SANS.CHR şrifti }
GothicFont	=4; {Gotiki GOTH.CHR şrifti }

Tekstiň ugruny görkezmek üçin aşakdaky hemişelikleri ulanmak mümkin:

Const

HorizDir

VertDir

Size parametri 1-den 10-a çenli aralykdan bitin bahany alyp bilýär.

SetTextJustify prosedurasy. Bu prosedura görkezijiniň ýerleşýän ýerine görä teksti nähili deňlemelidigini kesgitleýär. Sözbaşysy:

Procedure SetTextJustify(Horiz,Vert:Word);

Bu ýerde Horiz – kese deňlemegiň, Vert – dik deňlemegiň şertlerini görkezýär.

SetTextJustify prosedurasynyň parametrleri hökmünde aşakdaky hemişelikleri ulanmak bolýar:

const

LeftText	=0; {Görkeziji tekstden çepde }
CenterText	=1; { çepdenwe sagdan, aşakdan we ýokardan simmetrik }
RidhtText	=2; { Görkeziji tekstden sagda }
BottomText	=0; { Görkeziji tekstden aşakda }
TopText	=2; { Görkeziji tekstden ýokarda }

Aşakdaky programma dürli şriftleri görkezýär:

Uses Graph, CRT;

const

FontNames: **array** [1..10] **of String** [4] =(‘TRIP’,’LITT’
,’SANS’,’GOTH’, ‘SCRI’,
,’SIMP’,’TSCR’,’LCOM’,’EURO’,’BOLD’);

Tab1:= 50;

Tab2 := 150;

Tab3 :=220;

var

d,r,Err; Y,dY, Size, MaxFont,k: Integer;

NT,SizeT,Sy

mbT: **String**;

c: Char;

Procedure

OutTextWith

Tab(S1/S2,S

3,S4:**String**)

;

begin

MoveTo((Tab1-TextWidth(S1)) div2,Y);

OutText(S1) ;

MoveTo(Tab1+(Tab2-Tab1-TextWidth(S2)) **div** 2,Y);

OutText(S2);

MoveTo(Tab2+(Tab3-Tab2-TextWidth(S3)) **div** 2,Y) ;

OutText(S3) ;

if S4==‘Symbols’ then

MoveTo((Tab3+GetMaxX-TextWidth(S4)) div2,Y)

else MoveTo(Tab3+3,Y) ;

O

utTe

xt(S4

)

end;

```

begin
  InitGraph(d,r,''
  );
Err := GraphResult;
if Err<>grOk then
  WriteLn(GraphErrorMsg(
  Err)) else begin
    {$IFDEF VER70}
      MaxFont:=10;
    {$ELSE}
      MaxFont := 4;
    {$ENDIF}

    SetTex
      tStyle(
        1,0,4);
    Y := 0;
    OutTextWithTab('N','Name', 'Size', 'Symbols') ;
    y := 4*TextHeight('Z') div 3;
    Line(0,Y,GetMaxX,Y) ;
    Y := 3*TextHeight('Z') div 2;
    dY := (GetMaxY-Y) div (MaxFont);
    SymbT := '' ;
    for c := 'a' to 'z' do
      SymbT := SymbT+c;
    for k := 1 to
      MaxFont do begin
Size := 0;
repeat
inc(Size) ;
SetTextStyle(k,0,Size+1);
until (TextHeight ('Z')>=dY) or (Size=10)
or (TextWidth(FontNames[k])> (Tab2- Tab1));
Str(k,NT);
STR(Size,SizeT) ;

```

```

SetTextStyle(k, HorizDir, Size) ;
    OutTextWithTab(NT, FontNames[k], SizeT, SymbT) ;
inc(Y, dY)
end;
Rectangle(0,0,GetMaxX,GetMaxY) ;
Line(Tab1,0,Tab1,GetMaxY) ;
Line(Tab2,0,Tab2,GetMaxY);
Line(Tab3,0,Tab3,GetMaxY);
ReadLn;
CloseGraph
end end.

```

§26. Sagady programmirmek

DOS moduly kompýuterdäki sagat we sene bilen işlemäge, şeýle hem faýllaryň döredilen wagty we senesini hasaba almaga niýetlenen proseduralary saklaýar. Olaryň sanawy aşakdaky tablissada getirilýär.

Prosedura	Ýerine ýetirýän işi
GetDate(Var Yyl, Ay, Gun, HG : Word) SetDate(Yyl, Ay, Gun: Word)	Kompýuteriň sagadyndan ýyly, aýy, güni we hepdäň günini okaýar Kompýuteriň sagadyny düzedýär we onda ýyly, aýy, güni günini täzedden goýýar
GetTime(Var Sag, Min, Sek, Sek100:Word) SetTime(Sag, Min, Sek, Sek100:Word)	Kompýuteriň sagadyndan wagty okaýar Kompýuteriň sagadynda wagty düzedýär
PackTime (Var DT : DateTime; Var T : Longint) UnPackTime (T : Longint; Var DT :DateTime)	Faýla goýmak üçin senäniň we wagtyň ykjam ýazgysyny döredýär Faýldan oakalan senäni we wagty açýar
GetFTime (Var f; Var T : Longint) SetFTime (Var f; T : Longint)	Açylan f faýl üçin senäniň we wagtyň ykjam ýazgysyny okaýar Açylan f faýl üçin senäniň we wagtyň ykjam ýazgysyny goýýar

Indi bolsa ýokardaky tablissada getirilen proseduralara doly düşündirtiş bereliň.

GetDate we **SetDate** proseduralary kompýuteriň sagadyndan senäni okamaga we ony üýtgetmäge mümkinçilik berýär. GetDate prosedurasy öz parametrleriniň üsti bilen ýylyň, aýyň, günüň we hepdäniň günini berýär. SetDate prosedurasy bolsa, kompýuteriň sagadyndaky senäni üýtgetmeklige ýardam berýär. Tablissadan görnüşi ýaly SetDate prosedurasynda heşdäniň günini görkezýän **HG** parametri ýokdur. Onuň bahasyny SetDate prosedurasynyň özi awtomatiki hasaplaýar.

Senäniň girizilýän bahalaryna hem çäklendirmeler bardyr. Girizilýän ýyl 1980...1099 aralykda, aý 1...12 aralykda, gün 1...31 aralykda bolmalydyr. Şu görkezilen çäklerden çykylanda SetDate prosedurasy hasaba alynmaýar(ol ýerine ýetirilmän programmadaky indiki operatora geçilýär). GetDate we SetDate proseduralaryny gös göni niýetlenen işi üçin dälde, eýsem 1980...2099 ýyllaryň aralygyndaky islendik senedäki hepdäniň günini tapmak üçin hem ulanmak mümkindir(aşakdaky programma seret):

Uses DOS;

Type

DayString = String[11];

Const

Days : Array[1..7] of DayString = {hepdäniň günleri}

(‘Baş günü’; ‘Ýaş günü’; ‘Hoş günü’; ‘Sogap günü’;

‘Anna günü’; ‘Ruh günü’; ‘Dyňç günü’);

Function WhatDay(Y, A, G: Word): String;

Var

Y1, A1, G1, N1: Word;

BEGIN

GetDate(Y1, A1, G1, N1); {Kompýuteriň sagadyndaky senäni ýatda saklaýarys}

SetDate(Y, A, G); GetDate(Y, A, G, N1);

WhatDay:=Days[N1];

```

SetDate(Y1, A1, G1); {Kompýuteriň sagadyndaky öňki senäni
dikeldýäris}
END;
VAR
yyl, ay,gun:Word;
BEGIN
Write('Yyly girizin '); ReadLn(yyl);
Write('Ayy girizin '); ReadLn(Ay);
Write('Guni girizin '); ReadLn(Gun);
WriteLn;
WriteLn(gun:2,'/',ay:2,'/',Yyl:4,'→', WhatDay(yyl, ay,gun);
ReadLn;
END.

```

GetTime(Var Sag, Min, Sek, Sek100:Word) we SetTime(Sag, Min, Sek, Sek100:Word) proseduralary kompýuteriň içki sagadyndan wagty okamaga we ol sagady düzetmäge mümkinçilik berýärler. Sagady düzetmek üçin SetTime prosedurasy çagyrylanda parametrleri gös-göni san bilen hem bermek mümkindir. Goýulýan bahalar aşakdaky çäklerde bolmalydyr: sagat - 0... 23, minut - 0 ... 59, millisekunt - 0... 99 . Eger girizilýän bahalar degişli çäklerden çykýan bolsa, onda sagat üýtgedilmeýär. Aşakdaky programmada GetTime prosedurasy ulanylyp programmanyň işeyän wagtyny hasaplanýar:

```

Uses DOS;
Var
    h, min, s, ms : Word;
    Time : Real;
Begin
    GetTime(h, min, s, ms);

    Time:=(h*60+min)*60+s+ms/100;

    Write('ENTER klawişany basyn');

```

```

GetTime(h, min, s, ms);
Time:= (h*60+min)*60+s+ms/100 – Time;
ms:=Trunc(100*Frac(Time));
h:=Trunc(Time) div 3600;
min:=Trunc(Time-3600*h) mod 60;
s:=Trunc(Time-3600*h) mod 60;
WriteLn('Programmanyň aktiw işlän wagty ', h: 1, ' sagat',
min:1, ' minut', s:1,'sekunt',ms:1,' ms');
ReadLn;
end.

```


PORTLAR BILEN İŞLEMEK

§27. Porta ýazmak we portdan okamak

Turbo Paskalada öňünden kesgitlenen PORT we PORTW massiwleri bardyr. PORT massiwi BYTE, PORTW massiwi bolsa WORD tiplidir. Bu massiwler gös-göni porta ýazmaga we portdan okamaga mümkinçilik berýär. PORT ýa-da PORTW massiwiň elementine baha bermek porta ýazmaklygy amala aşyrýar, aňlatmanyň içinde ol massiwleriň elementlerini ulanmaklyk bolsa portdan okamaklyga getirýär.

Kesgitlenen PORT sözünüň yzyndan gelýän kwadrat skobkanyň içinde portyň adresi görkezilýär. Ol ardesi adatça onaltylyk sanawyş ulgamynda berilýär. Ol onluk hasaplaýyş ulgamynda hem görkezilip biliner. Mysal üçin PORT[\$378] ýazgy \$378 adresli porta ýüzlenmekligi aňladýar. Bu port parallel LPT port bolup durýar. Parallel porta adatça printerler birikdirilýär.

Aşakdaky programma kompýuteriň \$378 portyna 1 sany ýazmaklygy amala aşyrýar:

```
Begin  
PORT[$378]:=1;  
end.
```

Indiki programma bolsa kompýuteriň \$456 portyndan bir baýty okamaga mümkinçilik berýär:

```
Var  
a:Byte;  
Begin  
a:=Port[$456];  
write(a);  
end.
```

PORT we PORTW massiwleri ulanyp kompýuteriň portlaryny ulanyp, şol portlara dakylan daşky gurluşlary

dolandyrmak, ýa-da şol daşky gurluşlardan maglumatlary alyp, şol maglumatlary gaýtadan işlemek mümkin.

Aşakdaky programma okuw sagatlarynyň tertibi boýunça LPT porta (adresini \$378) signal iberýär. Ol signallary elektrik güýçlendirijiniň kömegi bilen güýçlendirip elektrik jaňynyň kakylmagyny gurnamak bolar. Bu programma okuw sapaklarynyň tertibinde gabat gelýän hepdäniň sanawjydygyny ýa-da maýdalawjydygyny hem kesgitlemäge mümkinçilik berýär. Programmany doly görnüşde getirýäris:

USES

Crt, Graph, Dos, EGAVGA, BGIFONT, BGIDRIV;

CONST

RSAG=80; RMIN=110; RSEK=120;

X0=320; Y0=240; R=170;

Fon=8;

VAR

h,m,s,hund,ho,h0,mo,so,XS,YS,XM,YM,XH,YH : word;

ha,ma,i1,i2,code1,code2,Gd,Gm : integer;

rdrob : 0..15;

drob : Char;

yyl,ay,gun,hep : word;

n : 1..4;

rrr : 1..15;

{-----}

Procedure D(R,Xo : word; Var X,Y : word);

Begin

X:=X0+Round(R*sin(Xo*Pi/180));

Y:=Y0-Round(R*cos(Xo*Pi/180));

End;

{----- SANAWJY_MAYDALAWJY -----}

Procedure San_May;

Begin

SetColor(Red);

SetLineStyle(SolidLn,0,3);

```

Line(50,100,150,100);

if (drob='S') then begin
  if rdrob=0 then rdrob:=10;
  if (rdrob=10) and (s mod 4 = 0) then rdrob:=0;
  SetColor(rdrob); OutTextXY(73,85,'SANAWJY');
  end;

if (drob='M') then begin
  if rdrob=0 then rdrob:=10;
  if (rdrob=10) and (s mod 4 = 0) then rdrob:=0;
  SetColor(rdrob); OutTextXY(60,110,'MAYDALAWJY');
OutTextXY(77,106,'^');
  end;

End;
{-----}
Procedure p0;
Var
  ay : array[1..12] of 1..31;
  y,m,d,w : word;
  s,s1,s2 : Longint;
  L : boolean;

{=====}
Function fL(k : word) : boolean;
Begin
  L:=(k mod 4 =0) xor ((k mod 100 = 0) and (k mod 400 <>
0)); Fl:=L;
End;

{=====}
Procedure p01(d,m,y : word; var s : longint);
var i : word;
Begin
  s:=0;

```

```

    for i:=1 to y-1 do
        if fl(i) then s:=s+366 else s:=s+365;
    if fl(y) then ay[2]:=29;
    for i:=1 to m-1 do s:=s+ay[i];
    s:=s+d;
End;
{=====}

```

```

Begin
    ay[1]:=31; ay[2]:=28; ay[3]:=31; ay[4]:=30; ay[5]:=31;
ay[6]:=30;
    ay[7]:=31; ay[8]:=31; ay[9]:=30; ay[10]:=31; ay[11]:=30;
ay[12]:=31;

```

```

    GetDate(y,m,d,w);
    if m<9 then p01(01,09,y-1,s1) else p01(01,09,y,s1);
    if (s1 mod 7 = 0) then drob:='M' else drob:='S';
    p01(d,m,y,s2);

```

```

    for s:=s1+1 to s2 do
        if (s mod 7 = 1) then
            if drob='S' then drob:='M' else drob:='S';

```

```

end;
{-----}

```

```

Procedure p1;

```

```

Var

```

```

    sec, R1 : Integer;

```

```

    X3,Y3 : word;

```

```

Begin

```

```

    setcolor(white);
    Circle(X0,Y0,R+10);
    Circle(X0,Y0,R);

```

```

SetFillStyle(SolidFill, 9);
FloodFill(X0+R+4,Y0,white);

SetFillStyle(SolidFill, Fon);
FloodFill(X0+8,Y0,white);

sec:=0;
While sec<=360 do
begin
    IF sec mod 30 = 0 then begin setcolor(yellow);
R1:=150 end
        else begin setcolor(white); R1:=163 end;
        D(R,sec,XS,YS); D(R1,sec,X3,Y3);
        Line(XS,YS,X3,Y3);
        sec:=sec+6;
    end;
End;
{-----}
Procedure p2;
Const SecColor=15; MinColor=14; SagColor=14;
Label 1,2;
Var
    sagat,sagato,minut,sekunt : word;
    XSO,YSO : word;
    hm,hl,hw,ht : word;
    hs,ms,ss,sss,sssm,sssl,sssw,ssst : string;
    yyls,guns : string;
    ach : 0..1;
    a : 0..15;
Begin
    so:=so-1; mo:=m; ho:=h0;
    ach:=0; XSO:=X0; YSO:=Y0; XM:=X0; YM:=Y0;
    XH:=X0; YH:=Y0;
    a:=0;
    sss:='      ';

```

```

                sssm:='      ' ; sssl:='      ' ;
                sssw:='      ' ; ssst:='      ' ;
REPEAT
    GetDate(yyl,ay,gun,hep);
    San_May;
    Bayramchylyklar;
    Doglan_gunler;
{ ***** }

    SetColor(11); SetTextStyle(1,0,1);{9}
    Outtextxy(380,110,'I');
    Outtextxy(425,160,'II');
    Outtextxy(445,228,'III');
    Outtextxy(425,300,'IV');
    Outtextxy(380,348,'V ');
    Outtextxy(310,368,'VI');
    Outtextxy(240,348,'VII');
    Outtextxy(194,300,'VIII');
    Outtextxy(175,228,'IX');
    Outtextxy(192,160,' X');
    Outtextxy(243,110,'XI');
    Outtextxy(310,90,'XII');

    setcolor(5); Circle(X0,Y0,5);
    str(yyl,yyls);
    setcolor(14);
    SetTextStyle(1,0,3); OutTextxy(293,145,yyls);
    if (yyl>1999) and (yyl<2100) then
        begin
            a:=a+1;
            if a>15 then a:=1;
            if a=Fon then a:=a+1;
            setcolor(a); SetTextStyle(1,0,3);
            OutTextxy(245,180,'ALTYN ASYR');
        end;

```

```

SetColor(13); SetTextStyle(0,0,0);
Case ay of
  1: begin Outtextxy(360,255,'Turkmenbasy');
    Outtextxy(367,248,chr(46));
      Outtextxy(371,248,chr(46));
    Outtextxy(434,262,chr(39)); end;
  2: begin Outtextxy(383,255,'Baydak');
    Outtextxy(402,253,chr(39)); end;
  3: Outtextxy(382,255,'Nowruz');
  4: Outtextxy(355,255,'Gurbansoltan');
  5: Outtextxy(365,255,'Magtymguly');
  6: Outtextxy(390,255,'Oguz');
  7: Outtextxy(382,255,'Gorkut');
  8: Outtextxy(366,255,'Alp Arslan');
  9: Outtextxy(378,255,'Ruhnama');
  10: begin Outtextxy(362,255,'Garassyzlyk');
    Outtextxy(395,262,chr(39)); end;
  11: Outtextxy(382,255,'Sanjar');
  12: Outtextxy(366,255,'Bitaraplyk');
end;

str(gun,guns);
if gun<9 then guns:='0'+guns;
if hep=0 then setcolor(12) else setcolor(2);
SetTextStyle(1,0,3); Outtextxy(395,225,guns);

SetColor(12); SetTextStyle(0,0,0);
Case hep of
  1: begin Outtextxy(375,270,'(Basgun)');
    OutTextXY(400,277,chr(39));
      OutTextxy(414,264,chr(46));
    OutTextxy(417,264,chr(46)); end;
  2: begin Outtextxy(375,270,'(Yasgun)');
    OutTextXY(400,277,chr(39));

```

```

        OutTextxy(414,264,chr(46));
OutTextxy(417,264,chr(46)); end;
3: begin Outtextxy(375,270,'(Hosgun)');
OutTextXY(400,277,chr(39));
        OutTextxy(414,264,chr(46));
OutTextxy(417,264,chr(46)); end;
4: begin Outtextxy(365,270,'(Sogapgun)');
OutTextXY(420,264,chr(46));
        OutTextXY(423,264,chr(46)); end;
5:   Outtextxy(384,270,'(Anna)');
6: begin Outtextxy(375,270,'(Ruhgun)');
        OutTextxy(414,264,chr(46));
OutTextxy(417,264,chr(46)); end;
0: begin Outtextxy(368,270,'(Dyncgun)');
OutTextXY(402,277,chr(39));
        OutTextxy(415,264,chr(46));
OutTextxy(418,264,chr(46)); end;
end;

        str(h,hs); str(m,ms); str(s,ss);
        if h<10 then hs:='0'+hs;
        if m<10 then ms:='0'+ms;
        if s<10 then ss:='0'+ss;
        SetTextStyle(0,0,1);
        setcolor(Fon); outtextxy(289,340,sss);
        sss:=hs+':'+ms+':'+ss;
        setcolor(13); outtextxy(289,340,sss);

{----- Moskwa wagty -----}
        SetColor(LightRed); outtextxy(20,425,'Moskwa
wagty:');
        if h=0 then hm:=22;
        if h=1 then hm:=23;
        if h>=2 then hm:=h-2;
        str(hm,hs); str(m,ms); str(s,ss);
        if hm<10 then hs:='0'+hs;

```



```

    if m<10 then ms:='0'+ms;
    if s<10 then ss:='0'+ss;
    SetTextStyle(0,0,1);
    setcolor(Black); outtextxy(160,425,sssm);
    sssm:=hs+':'+ms+':'+ss;
    setcolor(LightCyan); outtextxy(160,425,sssm);

{----- London wagty -----}
    SetColor(LightRed); outtextxy(20,435,'London
wagty:');
    if h=0 then hl:=19;
    if h=1 then hl:=20;
    if h=2 then hl:=21;
    if h=3 then hl:=22;
    if h=4 then hl:=23;
    if h>=5 then hl:=h-5;
    str(hl,hs); str(m,ms); str(s,ss);
    if hl<10 then hs:='0'+hs;
    if m<10 then ms:='0'+ms;
    if s<10 then ss:='0'+ss;
    SetTextStyle(0,0,1);
    setcolor(Black); outtextxy(160,435,sssl);
    sssl:=hs+':'+ms+':'+ss;
    setcolor(LightCyan); outtextxy(160,435,sssl);

{----- Wasington wagty -----}
    SetColor(LightRed); outtextxy(20,445,'Wasington
wagty:');
    outtextxy(37,451,chr(39));
    if h=0 then hw:=13;
    if h=1 then hw:=14;
    if h=2 then hw:=15;
    if h=3 then hw:=16;
    if h=4 then hw:=17;
    if h=5 then hw:=18;

```

```

if h=6 then hw:=19;
if h=7 then hw:=20;
if h=8 then hw:=21;
if h=9 then hw:=22;
if h=10 then hw:=23;
if h>=11 then hw:=h-11;
str(hw,hs); str(m,ms); str(s,ss);
if hw<10 then hs:='0'+hs;
if m<10 then ms:='0'+ms;
if s<10 then ss:='0'+ss;
SetTextStyle(0,0,1);
setcolor(Black); outtextxy(160,445,sssw);
sssw:=hs+'.'+ms+'.'+ss;
setcolor(LightCyan); outtextxy(160,445,sssw);

```

```

{ ----- Tokio wagty ----- }
SetColor(LightRed); outtextxy(20,455,'Tokio
wagty:');

```

```

if h<20 then ht:=h+4;
if h=20 then ht:=00;
if h=21 then ht:=01;
if h=22 then ht:=02;
if h=23 then ht:=03;
str(ht,hs); str(m,ms); str(s,ss);
if ht<10 then hs:='0'+hs;
if m<10 then ms:='0'+ms;
if s<10 then ss:='0'+ss;
SetTextStyle(0,0,1);
setcolor(Black); outtextxy(160,455,ssst);
ssst:=hs+'.'+ms+'.'+ss;
setcolor(LightCyan); outtextxy(160,455,ssst);

```

```

{ ----- SEKUNT ----- }
1:      GetTime(h,m,s,hund);
{      if (h=0) and (m=0) and (s=0) then p0;}

```

```

        SetLineStyle(SolidLn,0,3);
        h0:=h;
        if h0>12 then h0:=h0-12;
            sekunt:=6*s;
            D(RSEK,sekunt,XS,YS);
            if ach=0 then NoSound;
        if s<>so then begin
            if ach=0 then begin

                if (h=8) and (m=29) and (s>50) then begin Sound (2000);
                goto 2; end;
                if (h=9) and (m=49) and (s>50) then begin Sound (2000);
                goto 2; end;
                if (h=9) and (m=59) and (s>50) then begin Sound (2000);
                goto 2; end;
                if (h=11) and (m=19) and (s>50) then begin Sound (2000);
                goto 2; end;
                if (h=11) and (m=59) and (s>50) then begin Sound (2000);
                goto 2; end;
                if (h=13) and (m=19) and (s>50) then begin Sound (2000);
                goto 2; end;
                if (h=13) and (m=29) and (s>50) then begin Sound (2000);
                goto 2; end;
                if (h=14) and (m=49) and (s>50) then begin Sound (2000);
                goto 2; end;

                if (h=14) and (m=59) and (s>50) then begin Sound (2000);
                goto 2; end;
                if (h=16) and (m=19) and (s>50) then begin Sound (2000);
                goto 2; end;
                if (h=16) and (m=29) and (s>50) then begin Sound (2000);
                goto 2; end;
                if (h=17) and (m=49) and (s>50) then begin Sound (2000);
                goto 2; end;
            end;
        end;

```

```

    if (s>0) and (s<1) then goto 2;
    Sound(20);
2:
    end;
    so:=s;
    setcolor(Fon); Line(X0,Y0,XSO,YSO);
    XSO:=XS; YSO:=YS;
    end
    else begin
    setcolor(SecColor); Line(X0,Y0,XS,YS);
    goto 1;
    end;
{----- MINUT -----}
    SetLineStyle(SolidLn,0,3);
    minut:=6*m;
    D(RMIN,minut,XM,YM);
    if m<>mo then begin
    D(RMIN,6*mo,XM,YM);
    setcolor(Fon);
        Line(X0-1,Y0-1,XM-1,YM-1);
        Line(X0,Y0,XM,YM);
        Line(X0+1,Y0+1,XM+1,YM+1);
    mo:=m;
    D(RMIN,minut,XM,YM);
    setcolor(MinColor);
        Line(X0-1,Y0-1,XM-1,YM-1);
        Line(X0,Y0,XM,YM);
        Line(X0+1,Y0+1,XM+1,YM+1);
    end
    else
    begin
    setcolor(MinColor);
        Line(X0-1,Y0-1,XM-1,YM-1);
        Line(X0,Y0,XM,YM);
        Line(X0+1,Y0+1,XM+1,YM+1);

```

```

end;
{----- SAGAT -----}
SetLineStyle(SolidLn,0,3);
sagat:=(s+m*60+h0*60*60) div 120;
if sagat<>sagato then
begin
sagato:=sagat;
setcolor(Fon);
Line(X0-2,Y0-2,XH-2,YH-2);
Line(X0-1,Y0-1,XH-1,YH-1);
Line(X0,Y0,XH,YH);
Line(X0+1,Y0+1,XH+1,YH+1);
Line(X0+2,Y0+2,XH+2,YH+2);
D(RSAG,sagat,XH,YH);
setcolor(SagColor);
Line(X0-2,Y0-2,XH-2,YH-2);
Line(X0-1,Y0-1,XH-1,YH-1);
Line(X0,Y0,XH,YH);
Line(X0+1,Y0+1,XH+1,YH+1);
Line(X0+2,Y0+2,XH+2,YH+2);
end
else
begin
setcolor(SagColor);
Line(X0-2,Y0-2,XH-2,YH-2);
Line(X0-1,Y0-1,XH-1,YH-1);
Line(X0,Y0,XH,YH);
Line(X0+1,Y0+1,XH+1,YH+1);
Line(X0+2,Y0+2,XH+2,YH+2);
end;
{----- ALARM -----}
If (h=ha) and (m=ma) and (s<=9) then begin Sound(300*s);
ach:=1; end;
If (h=ha) and (m=ma) and (s=10) then begin NoSound;
ach:=0; end;

```

if (m=0) and (s=0) then begin Sound(4000); ach:=1; end;
if (m=0) and (s=1) then begin NoSound; ach:=0; end;

{ Port[\$378]:=\$05; operator LPT porta ikilik sanaýyş
ulgamyndaky 00000101 signaly iberyär. Portyň degişli
iňňejiklerinde 0 we 1 signallar emele gelýär.
'0' signal – 0 wolt, '1' – signal bolsa takmynan 5 wolt.
Şeýlelikde '1' signal adaty bipolýar tranzistorlary açmak üçin
ýeterlikdir}

If (h=08) and (m=30) and (s<10) then begin Port[\$378]:=\$05;
Sound(300*s); ach:=1; end;
If (h=08) and (m=30) and (s=10) then begin Port[\$378]:=\$00;
NoSound; ach:=0; end;
If (h=09) and (m=50) and (s<10) then begin Port[\$378]:=\$05;
Sound(300*s); ach:=1; end;
If (h=09) and (m=50) and (s=10) then begin Port[\$378]:=\$00;
NoSound; ach:=0; end;
If (h=10) and (m=00) and (s<10) then begin Port[\$378]:=\$05;
Sound(300*s); ach:=1; end;
If (h=10) and (m=00) and (s=10) then begin Port[\$378]:=\$00;
NoSound; ach:=0; end;
If (h=11) and (m=20) and (s<10) then begin Port[\$378]:=\$05;
Sound(300*s); ach:=1; end;
If (h=11) and (m=20) and (s=10) then begin Port[\$378]:=\$00;
NoSound; ach:=0; end;
If (h=12) and (m=00) and (s<10) then begin Port[\$378]:=\$05;
Sound(300*s); ach:=1; end;
If (h=12) and (m=00) and (s=10) then begin Port[\$378]:=\$00;
NoSound; ach:=0; end;
If (h=13) and (m=20) and (s<10) then begin Port[\$378]:=\$05;
Sound(300*s); ach:=1; end;
If (h=13) and (m=20) and (s=10) then begin Port[\$378]:=\$00;
NoSound; ach:=0; end;

```

    If (h=13) and (m=30) and (s<10) then begin Port[$378]:=$05;
    Sound(300*s); ach:=1; end;
    If (h=13) and (m=30) and (s=10) then begin Port[$378]:=$00;
    NoSound; ach:=0; end;
    If (h=14) and (m=50) and (s<10) then begin Port[$378]:=$05;
    Sound(300*s); ach:=1; end;
    If (h=14) and (m=50) and (s=10) then begin Port[$378]:=$00;
    NoSound; ach:=0; end;

    If (h=15) and (m=00) and (s<10) then begin Port[$378]:=$05;
    Sound(300*s); ach:=1; end;
    If (h=15) and (m=00) and (s=10) then begin Port[$378]:=$00;
    NoSound; ach:=0; end;
    If (h=16) and (m=20) and (s<10) then begin Port[$378]:=$05;
    Sound(300*s); ach:=1; end;
    If (h=16) and (m=20) and (s=10) then begin Port[$378]:=$00;
    NoSound; ach:=0; end;
    If (h=16) and (m=30) and (s<10) then begin Port[$378]:=$05;
    Sound(300*s); ach:=1; end;
    If (h=16) and (m=30) and (s=10) then begin Port[$378]:=$00;
    NoSound; ach:=0; end;
    If (h=17) and (m=50) and (s<10) then begin Port[$378]:=$05;
    Sound(300*s); ach:=1; end;
    If (h=17) and (m=50) and (s=10) then begin Port[$378]:=$00;
    NoSound; ach:=0; end;
    {-----}
    UNTIL 1=2;{KeyPressed; 1=2}
    NoSound;
    CloseGraph;
End;
{-----}
BEGIN
RANDOMIZE;
rdrob:=10;
n:=1; rrr:=15;

```

```

If (ParamStr(1)=") and (ParamStr(2)=") then begin
    ha:=0; ma:=0;
        end
        else
        begin
Val(ParamStr(1),i1,code1); Val(ParamStr(2),i2,code2);
If (code1=0) and (code2=0) then begin ha:=i1; ma:=i2; end;
        end;
Gd := Detect; InitGraph(Gd, Gm, "");
if GraphResult <> grOk then Halt(1);

SetCBreak(False);
p1;
p0;
p2;
END.

```


Edebiýatlar

1. Türkmenistanyň Konstitusiýasy. Aşgabat, 2008.
2. Gurbanguly Berdimuhamedow. Ösüşiň täze belentliklerine tarap. Saýlanan eserler. I tom. Aşgabat, 2008.
3. Gurbanguly Berdimuhamedow. Ösüşiň täze belentliklerine tarap. Saýlanan eserler. II tom. Aşgabat, 2009.
4. Gurbanguly Berdimuhamedow. Garaşsyzlyga guwanmak, Watany, Halky söýmek bagtdyr. Aşgabat, 2007.
5. Gurbanguly Berdimuhamedow. Türkmenistan – sagdynlygyň we ruhubelentligiň ýurdy. Aşgabat, 2007.
6. Türkmenistanyň Prezidenti Gurbanguly Berdimuhamedowyň Ministrler Kabinetiniň göçme mejlisinde sözlän sözi. (2009-njy ýylyň 12-nji iýuny). Aşgabat, 2009.
7. Türkmenistanyň Prezidentiniň «Obalaryň, şäherleriň, etrapdaky şäherçeleriň we etrap merkezleriniň ilatynyň durmuş-ýaşayyş şertlerini özgertmek boýunça 2020-nji ýyla çenli döwür üçin» Milli maksatnamasy. Aşgabat, 2007.
8. «Türkmenistany ykdysady, syýasy we medeni taýdan ösdürmegiň 2020-nji ýyla çenli döwür üçin Baş ugry» Milli maksatnamasy. «Türkmenistan» gazetiniň, 2003-nji ýylyň, 27-nji awgusty.
9. «Türkmenistanyň nebitgaz senagatyny ösdürmegiň 2030-njy ýyla çenli döwür üçin Maksatnamasy». Aşgabat, 2006.
10. Aşyralýew Ç. Kompýuter tehnologiýalary. Aşgabat, 2009 Kulyýew D. we başgalar. Kompýuterde işlemek. Aşgabat, 2005
11. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык ПАСКАЛЬ. М., 1988.
12. Поляков Д.Б., Кругов И.Ю. Программирование в среде ТУРБО ПАСКАЛЬ (версия 5.5) . М., 1992.
13. А. Шень. Программирование. Теоремы и задачи. М., 2004

14. Фрей Д. AutoCad 2007 и AutoCAD 2007 LT., Пресс – 2008
15. (Revised for the ISO Pascal Standart). Bern, 1984.

MAZMUNY

SÖZBAŞY	7
Birinji bölüm. Programmalaşdyrmakda kömekçi programmalaryň ulanylyşy	9
§1. Kömekçi programmalar. Kömekçi programmalaryň görnüşleri	9
§ 2. Formal parametrlr we parametrleriň iş ýüzündäki bahalary. Lokal we global parametrlr	11
§3. Rekursiýanyň ulanylyşy	15
§4. Kömekçi programmalary ulanmaklyga degişli mysallary çözmek	16
Ikinji bölüm. Programmalaşdyrmakda massiwleriň ulanylyşy	20
§5. Matrisalaryň üstündäki amallar	20
§6. Çyzykly deňlemeler sistemasyny çözmekde ulanylýan Gausyň usuluny programmalaşdyrmak.	25
§7. Kesgitleýjileri hasaplamagyň programmasyny düzmek	26
§8. Ters matrisany hasaplamagyň programmasyny düzmek	28
§9. Massiwleri tertiplemekdäki ulanylýan ýönekeýje usullar	29
Üçünji bölüm. Matematiki meseleleri programmalaşdyrmak	35
§10. Deňlemeleri çözmegiň ýakynlaşan usullary	35
§11. Kesgitli integraly hasaplamak	40
Dördünji bölüm. Senenamalar bilen balanyşykly meseleleri programmalaşdyrmak	45
§12. Senenamalar bilen balanyşykly meseleleri programmalaşdyrmak	45
§13. Baky kalendar programasy	46
§14. Baky kalendar programasy senenamalar bilen baglanyşykly dürli meseleleri çözmekde ulanmak	47
Bäşinji bölüm. Tekst faýllary bilen işlemek	52
§15. Simwollary we setirleri gaýtadan işlemeklige degişli meseleleri programmalaşdyrmak	
§16. Simwollaryň ASCII we ANSI kodlary. Conwertor programmasyny taýynlamak(ASCII ⇔ ANSI)	67
§17. Maglumatlary goramagyň usullary	70
§18. Telefon gözleg ulgamyny taýýarlamak	73
§19. Talypalaryň bilimine baha berýän testirleýji programmany taýýarlamak	80
Altynji bölüm. Oýunlary programmalaşdyrmak	83
§ 20. „X – O“ oýnuny programmalaşdyrmak	83
§21. „NIM“ oýnuny programmalaşdyrmak	87
§22. Küşt oýny bilen baglanyşykly meseleleri	97

programmalaşdyrmak	
Ýedinji bölüm. Grafiki ekrany dolandyrmak	100
§ 23. Displeý adapterleriniň grafiki iş kadalarynyň gysgaça häsiýetnamasy	100
§24. Grafiki kadada işlemek üçin ulanylýan proseduralar we funksiýalar. Figuralary hereketlendirmek	102
§25. Grafiki ekrana tekst çykarmak	110
§26. Sagady programmirmek	114
Sekizinji bölüm. Portlar Bilen İşlemek	118
§27. Porta ýazmak we portdan okamak	118
Edebiýatlar	134