

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Ш.Е.ЕСЕНОВ АТЫНДАҒЫ КАСПИЙ МЕМЛЕКЕТТІК ТЕХНОЛОГИЯЛАР ЖӘНЕ
ИНЖИНИРИНГ УНИВЕРСИТЕТІ
ПЕДАГОГИКАЛЫҚ ТЕХНОЛОГИЯЛАР ИНСТИТУТЫ

Урбисинова Г.О.

«Бағдарламалау пәнінен зертханалық жұмыстарды орындау»

050111- Информатика мамандықтарының студенттеріне арналған оқу-әдістемелік нұсқау

АҚТАУ-2010

ӘОЖ 004.3 (072)

Құрастырған: Урбисина Г.О..

Бағдарламалау пәнінен зертханалық жұмыстарды орындау:

Оқу-әдістемелік нұсқау. – Ақтау қ.. КМТЖИУ, 2010, 56 бет.

Пікір жазған: Педагогика ғылымдарының кандидаты С.Н. Идрисов

Бұл оқу-әдістемелік нұсқауда кең тараған Turbo Pascal тілінің бағдарламалау техникасы және осы тілді меңгеруге арналған зертханалық жұмыстардың жинағы берілген.

«Жүйелік бағдарламалау», «Бағдарламалау тілдері» пәндерін оқитын «Информатика» мамандығы студенттеріне , сонымен қатар осы бағдарламалау тілін меңгерушілер үшін өзіндік оқуға арналған оқу-әдістемелік нұсқау.

Ш.Есенов атындағы Каспий Мемлекеттік Технологиялар және Инжиниринг Университетінің Педагогикалық Технологиялар институтының оқу-әдістемелік кеңесінің шешімімен баспаға ұсынылды.

Ш. Есенов атындағы Каспий мемлекеттік технологиялар және инжиниринг университеті, 2010

Мазмұны

1. №1. Зертханалық жұмыс.Массивтер.....	3
2. №2. Зертханалық жұмыс.Екі өлшемді массивтер.....	12
3. №3. Зертханалық жұмыс. Жолдық қатарлар.....	16
4. №4. Зертханалық жұмыс. Жиындар.....	23
5. №5. Зертханалық жұмыс.Жазулар.....	27
6. №6. Зертханалық жұмыс.Көмекші программалар.....	34
7. №7. Зертханалық жұмыс.Функциялар.....	41
8. №8. Зертханалық жұмыс.Файлдар.....	46
9. Пайдаланған әдебиеттер.....	55

№1 Зертханалық жұмыс

Тақырыбы: Массивтер

Мақсаты:

Массивтер ұғымымен танысу. Массивтерді сипаттау. Массив элементтерін ендіру және шығару. Массив элементтеріне қолданылатын амалдарды пайдалана отырып программалар құра білу.

Теориялық түсініктемелер

Элементтерінің саны алдын-ала белгілі бір типті индексті айнымалылардың жиынтығын массив деп атайды. Массивтің типі күрделі болып табылады. Массивтерді программада қолдану үшін, оларға белгілі бір ат беріледі. Содан соң программаның VAR немесе TYPE бөлімінде ARRAY қызметші сөзімен сипаттап, компьютердің жадынан массив элементтеріне орын бөледі. Жалпы жазылу түрі төмендегідей:

TYPE “массив атауы” = ARRAY [X 1] OF X1;

VAR “массив атауы” : ARRAY [X 1] OF X2;

мұндағы:

“массив атауы” – массивті белгілеген идентификатор, X, A, MAS т.б.

ARRAY – массив ұғымын білдіретін қызметші сөз.

OF – қызметші сөз (бұл жерде жалғауларын білдіреді, мысалы бүтін шамалардан)

X1 – массив индексінің типі, нақтыдан (Real) басқа барлық типтер қолданылады.

X2 – массив элементінің негіздік типі.

Программада массивтерді сипаттаудың бірнеше жолдары бар. Алдымен типтер бөлімінде массивті бір идентификатор арқылы сипаттап алып, кейін айнымалылар бөлімінде массив атауын сол идентификатор арқылы сипаттауға болады. Сондай-ақ бірден айнымалылар бөлімінде де массивті толық сипаттап кетуге де болады.

Егер программада бірнеше массивтің индекстері мен негіздік типтері бірдей болса, онда оларды бір тізімге біріктіріп, жазып сипаттауға болады, төмендегі мысалдарда элементтері нақты сан болатын A, B, C, массивтерінің бәріне де компьютердің жадынан 100 элемент бөлінген, сондықтан олар бір тізімге біріктіріліп сипатталған.

Массивтердің типтер бөлімінде сипатталуы:

```
CONST N = 100; {N – элементтер саны}
```

```
TYPE Mas = Array [1..N] Of Real;
```

```
Element = (Ge, C, O, N, Na, K, Li, Cu, Zn);
```

```
Gaz Array [Ge .. N] Of Integer;
```

```
VAR A, B, C: Mas; X: Gaz;
```

Массивтердің айнымалылар бөлімінде сипатталуы:

```
CONST N = 100;
```

```
VAR A, B, C: Array [1..N] Of Real;
```

```
    D, E: Array [1..5] Of Char;
```

```
    Mezgl: Array [1..10] Of String;
```

```
    Y, Z: Array [1..20] Of Byte;
```

```
    R: Array ['A'..'Z'] Of Char;
```

```
    P: Array [1..10] Of String;
```

Жоғарыда сипатталған массивтердің барлығының да элементтерінің бір индексі бар. Бұндай массивтерді бір өлшемді массивтер дейді.

Массивтермен амалдарды орындау

Массивтердің өзімен жұмыс істеу үшін массивтің өз атауы қолданылады. Массивтің өзі тек қана « тең » , « тең емес » және меншіктеу амалдарына ғана қатыса алады. Бұл амалдарға

қатысатын массивтердің типтері мен элементтер саны бірдей болуы керек. Мысалы мынадай массив берілсе :

```
VAR X, Y: Array [1..5] Of Byte ;
```

онда бұл массивтерге келесі амалдар қолданғанда :

$X=Y$ - егер массивтердің индекстері сәйкес элементтері тең болса, онда нәтиже «ақиқат» болады (True).

$X < > Y$ - егер массивтердің индекстері сәйкес элементтерінің кемінде біреуі тең болмаса , онда нәтиже « ақиқат » болады (True).

$X:=Y$ – X массиві Y массивінің барлық элементтерін меншіктейді, ал Y массивінің элементтері өзгеріссіз қалады.

Массив элементтерімен амалдар орындау.

Массивті сипаттап болғаннан кейін, операторлар бөлімінде массив элементтеріне жай айнымалылармен орындалатын әртүрлі амалдар қолдануға болады. Оларды әр түрлі өрнектерде операндалар орнына, For, While, Repeat операторларының параметрлерінде, логикалық амалдарда салыстыру элементі ретінде пайдалануға болады. Программалық тексте массивтің әр элементі массивтің аты мен индексі арқылы жазылады. Элементтер индекстің мәндері арқылы реттеледі. Бір ескеретіні «индекс» және «индекстің типі» түсініктерін араластыруға болмайды. «Индекстің типі» түсінігі тек массивті сипаттайтын бөліктерде ғана қолданылады. Ал «индекс» түсінігі операторлар бөлігінде массив элементтерін белгілегенде қолданылады. Индекс ретінде әртүрлі өрнектер қолдануға болады. Көбінесе өрнектердің қарапайым бір түрі ретінде тұрақтылар мен айнымалылар қолданылады. Сондықтан да массив элементтерін басқаша индексті айнымалылар деп атайды. Егер математикада индекс жай жақшаға алынып немесе төменгі регистрде жазылса, Паскаль программаларында массив индекстері тік жақшаларға алынып жазылады.

Массив элементтері программадағы өрнектер мен меншіктеу операторларының сол жағында да, оң жағында да тұруы мүмкін. Массив элементтерінің негіздік типіне қолдануға болатын амалдарды массив элементтеріне де қолдануға болады, мысалы :

$A(51, -15, 5.5, -8.1, 15, 0, -7)$ –A массиві 7элементтен тұрады, элементтердің программада сипатталуы мен жазылуы :

```
VAR A: ARRAY [1..7] Of Real;
```

```
A [1] =51; A [2] =-15; A [3] =5.5; A [4] =-8.1; A [5] =15; A [6] = 0; A [7] = -7.
```

Массив элементтерінің индекстері өрнектер мен операторларда тік жақшаларға алынып жазылады.

```
Y:=5*A[1]+SQR(A[2]+A[3]); X:=COS A[5]+Sin(A[5]);
```

```
Read (A[k]) ; Writen(A[k] ) A[k] :=A[k+1];
```

```
If A[k]<5 Then S :=S+A[k];
```

```
While A[k] >=80 do Begin S:=S+ A[k] ; k :=k+1 End ;
```

Массив элементтерін ендіру және шығару.

Массив элементтерін ендіру және шығару цикл арқылы жүзеге асырылады. Егер элементтерді ендіру үшін Read операторы қолданылса, онда элементтер бір қатарға жазылып ендіріледі. Ал Readln операторы қолданылса, онда әр элементті жаңа қатардан енгізу керек.

```
.....
```

```
Readln (N);
```

```
For K: =1To N do Read (A[k]) ;
```

```
.....
```

```
Readln (N); K: =1;
```

```
Repeat Read (A [K]); K: =K+1
```

Until K>N;

```
.....  
Readln (N); K: =1;  
While K=<=N Do Begin  
Readln (A [K]); K: =K+1 End;
```

Программаның орындалу нәтижесінде N элементтен тұратын массив элементтерін клавиатурадан ендіруді ұйымдастырады, программа циклді оқығанда Read операторын N рет оқып тоқтап, массив элементтері мәндерін қашан клавиатурадан енгізгенше күтіп тұрады, бұл жерде K айнымалысы цикл параметрі қызметімен қоса массив индексі ретінде де қолданылып тұр.

Кейде программаның әр орындалуында массив элементтерінің мәндерін қайта-қайта клавиатурадан енгізу керек болады, бұл әсіресе енгізілетін элементтер саны өте көп болған кездерде программа орындаушы кісіні жалықтырып жібереді, сондықтан бұндай жағдайларда оларды бірден тұрақтылар (CONST) бөлімінде енгізіп қойса да болады:

```
CONST N=5 ; A: ARRAY [1..N] OF REAL =(40.5, 48,47.1, 52.3, 11.2);  
B : ARRAY [1..9] OF REAL=(31, 25, 40, 1.5, 48, 54.7, 12.3, 54.5, 21);
```

Массив элементтерін шығару процесі де ендіруге ұқсайды. Егер цикл денесінде элементтерді шығару үшін Write операторы қолданылса, онда массив элементтері экранға бір қатарға бір – біріне жабысып жазылып шығады. Бұл жағдайда нәтиженің көрнекілігін арттыру үшін Write операторында массивтен кейін бос орын жазылады. Ал Writeln операторы қолданылса, онда массивтің әр элементті жаңа қатарға жазылып шығады.

```
.....  
For K: =1 To N Do Write ('A[', K', ] =', A[K], ', ' );
```

```
.....  
K:=1  
Repeat Writeln (A [K]); K: =K+1  
Until K>N;
```

```
.....  
K:=1  
While K=<=N do  
Begin Write (A[K], ', ' ); K: = K+1 End;
```

Программаның орындалу нәтижесінде N элементтен тұратын массив элементтері экранға шығады, бұл жерде де K айнымалысы цикл параметрі қызметімен қоса массив индексі ретінде де қолданылып тұр.

Бір өлшемді массивтерге программа құру

1. Массивтерді көшіру.

А және В массивтері берілген. А массивіндегі элементтерді В массивіне көшіру программасын құру.

Бұл программаны екі түрлі жолмен құруға болады, бір массивке басқа бір массивтің элементтерінің мәндерін көшіру үшін төмендегідей операторларды қолдануға болады :

а) B: =A; - В массиві А массивінің барлық элементтерін меншіктейді.

ә) For K: =1 To N do B[K]:=A[K]; - циклдің әр орындалуында В массивінің К нөмірлі элементі А массивінің К нөмірлі элементінің мәнін меншіктейді, мұндағы К- цикл параметрімен бірге массив индексінің қызметін қоса атқарады.

Екі жағдайда да А массивінің элементтерінің мәні өзгермейді, ал В массивінің әрбір элементі А массивінен өзіне индексі сәйкес элементтің мәнін қабылдайды.

а) жай меншіктеу операторымен :

```
PROGRAM Kosh1;
USES Crt;
VAR K: Integer; A, B: Array [1..100] of Real;
BEGIN   Clrscr;
For K: =1 To 12 Do Readln (A [K], B[k]);
B: = A; Writeln;
For K: =1 To 12 Do Write (B[k]:6:1);
Readln  END.
```

ә) циклдік оператормен :

```
PROGRAM Kosh2;
VAR K: Integer; X, Y: Array [1..100] of Real;
BEGIN   For K: =1To 12 Do Readln (X [K], Y[k]);
For K: =1 To 12 Do B[k]:= A[k];
Writeln; For K: =1To 12 Do Write (B[k]:6:1);
Readln  END.
```

Массив элементтерінің қосындысын табу

Егер массив элементтерінің қосындысын **S** идентификаторымен белгілесек, онда қосынды табатын формула мынадай болады :

$$S: =X [1] +X[2]+X[3] +...+X[N] ;$$

Мұндағы : X-массив элементтері , N-массивтегі элементтер саны. Қосындыны сызықты программа құрып, жай меншіктеу операторы арқылы жоғарыдағы формуламен шығара салуға болады. Бірақ массив өте көп элементтерден тұрса , бұл формуланы ұзақ жазуға тура келеді. Сондықтан бұл формуланы циклдің ішіне кіргізіп қосындыны төмендегі формуламен есептеп шығару ыңғайлы болады :

FOR K : =1TO N DO S:=S +X[K];

Мұндағы :

K- цикл параметрі және массив индексінің қызметін бірге атқарып тұр.

N – элементтер саны .

өрнектің сол жағында тұрған **S** –қосынды

өрнектің оң жағында тұрған **S** –қосындының бастапқы (K=K-1 болғандағы) мәні, цикл басталмай тұрғандағы мәні 0-ге тең.

X[K] – индексі K мәніне тең массив элементі (мысалы K=1 болғанда **X [1]**).

2. X массивінің элементтерінің қосындысын табу программасын құру.

```
PROGRAM Kos1;
CONST
X: Array [1..9] Of Real = (12.5, 10.4, -14, 12.5, 0, 14.1, 15.7, 10, 8.5);
VAR K: Byte; S: Real;
BEGIN {экранға массив элементтері шығару циклі}
For K: =1 To 9 Do Write(X [K]: 6:1);
S: =0; {цикл басталмай тұрғандағы қосынды мәні}
For K: =1 To 9 Do S: = S+X [K]; {массив элементтерін қосу циклі}
Writeln; Writeln ('Массив эл.қосындысы =', S: 4:2);
Readln  END.
```

Программаның орындалу барысы :

1. цикл басталғанда цикл параметрі $k = 1$ мәнін қабылдайды да цикл денесіне барып k -ның осы мәніндегі қосындыны есептейді $S = S + X[k] = 0 + 12.5 \cdot k = 12.5$;
2. цикл параметрі келесі $k = 2$ мәнін қабылдап, цикл денесіне барып қосындыны есептейді $S = S + X[k] = 12.5 + 10.4 = 22.9$;
3. цикл параметрі келесі $k = 3$ мәнін қабылдап, цикл денесіне барып қосындыны есептейді $S = S + k = 22.9 + (-14) = 8.9$;
4. әрі қарай есептеу процесі дәл осылай циклдің орындалуы біткенше (k -ның мәні 9-дан асып кеткенше) жалғаса береді де, нәтижесінде цикл аяқталған соң S идентификаторының ең соңғы қабылдаған мәні массив элементтерінің қосындысын береді.

Дәл осы есепті басқа да циклдік операторлардың көмегімен шығаруға болады :

а) Repeat операторымен бұл программаны төмендегідей құруға болады:

```
PROGRAM Kos2;
CONST
X: Array [1..9] Of Real = (12.5, 10.4, -14, 12.5, 0, 14.1, 15.7, 10, 8.5);
VAR K: Byte; S: Real;
BEGIN For K: =1 To 9 Do Write(X [K]: 6:1); S: =0; K: =1
Repeat S:= S+X [K]; K: =K+1; Until K>9;
Writeln; Writeln ('Массив эл.қосындысы =', S: 4:2); Readln END.
```

ә) While операторымен бұл программаны төмендегідей құруға болады:

```
PROGRAM Kos2;
CONST
X: Array [1..9] Of Real = (12.5, 10.4, -14, 12.5, 0, 14.1, 15.7, 10, 8.5);
VAR K: Byte; S: Real;
BEGIN For K: =1 To 9 Do Write(X [K]: 6:1); S: =0; K: =1
While K<9 Do Begin S:=S+X[K] ; K:=K+1; End ; Writeln ;
Writeln ('Массив эл.қосындысы =', S: 4:2); Readln END.
```

Массив элементтерін шығару.

Кейбір есептеулерде белгілі бір шарттың орындалуына байланысты массив элементтерін іздеп табу керек болады. Ол үшін цикл арқылы массивтің барлық элементтерін берілген шамамен салыстыру процесін ұйымдастыру керек. Егер циклдің әр орындалуында массивтің жаңа элементі берілген шартты қанағаттандырса, онда бұл элемент ізделінген элемент болып табылады. Мысалы :

1. Массивтің 5-ке тең элементтерін экранға шығару :
..... **If A [K] =5 Then Write (A [K], ' ');**
2. Массивтің жұп элементтерінің нөмірлерін (индекстерін) шығару:
..... **If A [K] =5 Mod 2=0 Then Write ('K=', K, ' ');**
3. Массивтің 7-ден үлкен және 15-тен кіші элементтерінің санын табу:
N:=0 ; If (A [K] >7) And (A [K] <15) Then N: =N+1;
Writeln ('N=', N);
4. массивтің 10-нан үлкен элементтерінің қосындысын табу:
S: =0; If A [K]> 10 Then S: =S+A [K];
Writeln ('S=', S);
5. Массивтің 5-тен кіші элементтерінің көбейтіндісін табу:
P: =1; If A[K] <5 Then P:=P*A[K];
Writeln ('P=', P);

3. X(7) массивіндегі теріс элементтер санын табу.

```
PROGRAM Santabu;
VAR K, S: Integer; X: Array [1..100] Of Real;
BEGIN For K: =1 To 7 Do Read (X [K]);
S:=0 ;
For K: =1 To 7 Do If X [K] <0 Then S: =S+1;
  Writeln ('Теріс эл.саны =', S);
Readln END.
```

4. X(7) массивіндегі 0-ге тең элементтер индексіні шығару.

```
PROGRAM Indextabu;
CONST X : Array [1..7] Of Real =( 5.25, 14.2, 0,0,14.2,75 );
VAR K: Integer;
BEGIN K: =1;
While K=<7 Do Begin
IF X[K]=0 Then Writeln (' 0-ге тең эл.индексі =',K); Inc(K); End
Readln END.
```

5. X(7) массивіндегі тақ индексті элементтерді экранға шығару.

```
PROGRAM Takindex;
TYPE Mas = Array [1...100] Of Integer;
VAR K: Integer; X: Mas;
BEGIN For K: =1To 10 Do Read (X [K]);
Until K>10;
Readln; Readln END.
```

6. Жылдық, жаз айларының, қыс айларының, көктем және күз айларының орта температурасын шығару.

```
PROGRAM Ortatemp ;
CONST
X: Array [1..12] OF Real = (-12, -5, -1, 8, 14, 21, 37, 39, 32, 19, 13, 4);
VAR K: Integer;
    S, S1, S2, S3: Real;
BEGIN
For K: =1 To 12 Do Write (X [K]:4:1); Writeln;
S:=0; For K: =1 To 12 Do S: =S+X[K] ;
S1:=0; For K: =6 To 8 Do S1: =S1+X [K];
S2:=0; For K: =1 To 12 Do If (K<3) Or (K>11) Then S2:=S2+X [K];
S3:=0; For K: =3 To 11 Do If (K<6) Or (K>8) Then S3:=S3+X [K];
Writeln ('жылдық орта температура=', S: 12:4:1);
Writeln ('жаздық орта температура=', S1/3:4:1);
Writeln ('қыстық орта температура=', S2/3:4:1);
Writeln ('күздік –көктемдік орта температура=', S3/6:4:1);
Readln END.
```

7. A(10) массивіндегі теріс элементтерден жаңа массив құру.

```
PROGRAM Teryselement ;
CONST
A: Array [1..10] Of Integer = ( 20, 0, 0, -10, 20, 10, -11, -5, -5, 10 );
```

```

VAR K, N: Integer;
    B: Array [1..10] Of Integer;
BEGIN
For K: =1 To 10 Do Write (A [K]:4); Writeln;
N: =0;
For K: =1To 10 Do
If A [K]<0 Then Begin N:=N+1; B[N]:=A[K]; End;
Writeln ('жаңа массив ');
For K: =1 To N Do Write (B [K]:4);
Readln END.

```

Массив элементтерінің орнын ауыстыру.

8. А массивінің 1-ші 5-ші элементтерінің мәндерінің орнын ауыстыру.

```

PROGRAM Austr ;
USES Crt;
CONST A: Array [1..9] Of Real = (12.5, 10.4, -14, 2.5, 0, -4, 1.7, -7, 5 );
VAR K: Integer; B: Real
BEGIN Clrscr; For K: =1 To 9 Do Write (A [K]:6:1);
B: =A [5]; A [5]: =A [1]; A [1]:=B;
Writeln ('Жаңа массив элементтері: ');
For K: =1 To 9 Do Write (A[k]:6:1);
Readln END.

```

Массив элементтері мәндерінің орнын ауыстыру есептерін шығару үшін, берілген массивпен бір типтес қосымша айнымалы енгізіп, меншіктеу операторы арқылы элементтердің мәндері төмендегідей ауыстырылады :

B: =A [5]- массивтің 5-ші элементінің мәнін компьютердің жадында ұстап тұрады.
A [5]: = A [1]-массивтің 5-ші элементі 1-ші элементтің мәнін қабылдайды.
For K: =1 To 9 Do Write (A[K]:6:1)-жаңа элементтерінің мәндері ауыстырылған массив шығарады.

Массивтің ең үлкен не кіші элементтерін табу.

9. N элементтен тұратын А массивінің ең үлкен элементін табу программасын құру.

```

PROGRAM Max1;
CONST N =10;
VAR A: Array [1..N] Of Real; P: Real; K: Integer;
BEGIN For K: =1 To N Do Read (A [K]);
P: =A [1];
For K: =2 To N Do IF A [K] >P Then P: =A [K];
Writeln ('Массивтің ең үлкен элементті Max=', P: 4:2);
Readln END.

```

Max1 программасында For операторымен құрылған цикл массивтің ең үлкен элементін төмендегідей жолдармен табады:

1. Программаға қосымша типі А массивімен бірдей Р идентификаторы енгізіліп, ол VAR бөлімінде сипатталады.
2. Р идентификаторына цикл басталмай тұрып массивтің алғашқы элементі меншіктеліп (**P: =A [1]**), ол шартты түрде массивтің ең үлкені деп алынады.

3. **For K: =2 To N Do IF A [K] >P Then P: =A [K];** салыстыру циклі массивтің екінші элементінен бастап барлық элементтерді шартты үлкен элемент P-мен салыстырып, массивтің ең үлкен элементін шығарады. Егер циклдің кезекті орындалуында шарт орындалса, онда P идентификаторы – индексі циклдің сол орындалуындағы параметрге тең массив элементінің (**A[k]**) мәнін қабылдайды т.с.с. процестер цикл аяқталғанша жүреді. Цикл толық аяқталғаннан кейін массивтің ең үлкен элементі P идентификаторының соңғы мәніне тең болады.

10. A(10) массивінің ең кіші элементін тауып бірінші элементпен орын ауыстыру.

```
PROGRAM Elementmin;
CONST
  A: Array [1..10] Of Integer= (20, 10, 0, -10, 0, 0, -11,-5, -5, 0);
  VAR K, N, S: Integer;
BEGIN For K: =1 To 10 Do Write (A [K]:4); Writeln;
N:=A [K]; S:=1
For K: =2 To 10 Do IF A [k] <N Then Begin N:=A[K]; S:=K End;
A[S]:=A[1]; A[1]:=N; Writeln ( ' Жаңа массив: ');
For K: =1 To 10 Do Write ( A [K] :4);
Readln END.
```

11. $Y=X^3+5*X$ функциясының ең кіші мәнін табу.

```
PROGRAM Tenelement;
CONST
  X: Array [1..10] Of Real = (0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2);
  VAR K: Integer; Y:Array [1..11] Of Real; N:Real;
BEGIN Clrscr;
For K: =1 To 10 Do Write (X[K]:5:1); Writeln;
For K: =1 To 10 Do Y[K] :=Sgr (X[K])*X[K]-5*X[K];
N:= Y[1];
For K: =2 To 10 Do If Y[K] <N Then N:=Y[K];
Writeln ('min Y=', N: 4:1);
Readln END.
```

Массив элементтерін сұрыптау.

12 . Массив элементтерін кему бойынша сұрыптау.

```
PROGRAM Kemu1;
VAR K, I, S, P: Integer;
  X: Array [1..10] Of Integer;
BEGIN For K: =1 To 10 Do Read (X[K]); Writeln;
For K: =1 To 9 Do
For I:= K+1 To 10 Do
If X[I]> X[K] Then Begin
P:=X[K]; X[K]:=X[I]; X[I]:=P End;
Writeln; For K:= 1 To 10 Do Write (X[K]:5);
Readln; Readln END.
```

Kemu1 программасы массив элементтерін төмендегідей кему бойынша реттеп шығарады:

- 1.Программаға қосымша P идентификаторы ендіріліп, сипатталады, оның типі массивпен бірдей болуы тиіс.
2. Цикл ішінде цикл ұйымдастырылады:

Егер массивтегі элементтер санын N деп алсақ, онда сыртқы цикл параметрлерін 1-ден $(N-1)$ -дейін (For K:=1 To 9 Do), ал ішкі цикл параметрлері –сыртқы цикл индексінен кейінгі индекстен бастап N -ге дейін алынады

(For I:= K+1 To 10 Do).

3.Ішкі циклдегі шарт орындалса, онда сол шарттағы салыстырылған элементтердің орны ауыстырылады (If $X[I]>X[K]$ Then Begin P:=X[K]; X[K]:=X[I]; X[I]:=P End;).

4. Цикл ішіндегі циклдің орындалу нәтижесінде жаңа кему бойынша реттелген массив алынып, экранға шығарылады (For K:= 1 To 10 Do Write (X[K]:5;).

Массив элементтерін өсу бойынша реттеп шығару үшін шарттағы таңбаны

(If $X[I]<X[K]$) өзгерте салса жеткілікті.

Кездейсоқ сандардан тұратын массивтер.

13. Кездейсоқ сандардан тұратын массив элементтерін шығару.

```
PROGRAM Kezdeisok;
```

```
VAR K: Integer;
```

```
X: Array [1..9] Of Integer;
```

```
BEGIN Randomize;
```

```
{Программаның әр орындалуында Random функциясы әр түрлі мән шығару үшін қолданылатын процедура}
```

```
For K:=1 To 9 Do
```

```
X [K]:=Random (100) +1;
```

```
{Массивтің әр элементі 1-ден 100-ге дейінгі кез-келген кездейсоқ сан меншіктейді}
```

```
For K:=1 To 9 Do Write(X [K]:4);
```

```
Readln END.
```

Бақылау сұрақтары мен тапсырмалар.

1. Массивтер ұғымы, олардың программада сипатталуы.
2. Бір өлшемді массив элементтерін ендіру және шығару.
3. Бір өлшемді массив элементтерін реттеу.
4. Массивтердің типтері бөлімінде сипатталуы.
5. Массивтердің айнымалылар бөлімінде сипатталуы.
6. В(5) массиві берілген. Тақ нөмерлі элементтерінің квадраттарының қосындысын табу программасын құрыңыз.
7. В(7) сандық массиві берілген. Массивтің барлық жұп нөмерлі элементтерінің түбірлерінің мәнін экранға шығару программасын құрыңыз.
8. В(11) сандық массивінің бір – біріне ұқсамайтын элементтерінің санын табыңыз.
9. В(17) сандық массивіндегі бір – біріне ұқсас элементтерінің санын табыңыз.
10. А(10) массиві берілген. Экранға массивтің 15-тен үлкен 50-ден кіші элементтерінің нөмерін шығарыңыз. Ендіру циклін While операторымен, ал шығару циклін For операторымен құрыңыз.
11. А(10) массиві берілген. Экранға массивтің 3-ке қалдықсыз бөлінетін элементтерін шығарыңыз. Ендіру циклін For операторымен, ал шығару циклін While операторымен құрыңыз.
12. А(10) массиві берілген. 15-тен үлкен немесе -5-тен кіші болатын элементтер санын табыңыз. Ендіру және шығару циклдерін While операторымен құрыңыз.
13. А(10) массиві берілген. Массивтің оң элементтерінің қосындысының квадратын табыңыз. Ендіру циклін Repeat операторымен, ал шығару циклін While операторымен құрыңыз.
14. А(7) массиві берілген. Массивтің 0-ге тең элементтерді 5-пен ауыстыру программасын құрыңыз. Ендіруде – For, шығаруда - While циклдерін пайдаланыңыз.

№2 Зертханалық жұмыс

Тақырыбы: Екі өлшемді массивтер

Мақсаты:

Екі өлшемді массивтерді сипаттау. Массив элементтерін ендіру және шығару. Массив элементтеріне қолданылатын амалдарды пайдалана отырып программалар құра білу.

Теориялық түсініктемелер

Екі өлшемді массивтер.

Есептеулерде бір өлшемді массивтермен қатар көп индексті массивтер де кездеседі. Оларды индексінің санына қарай n – өлшемді массивтер деп атайды. Солардың ішінде Паскаль программаларында екі индексті массивтер кеңінен қолданылады. Оларды көбінесе матрицалар деп атайды. Матрица элементтерінің бірінші индексі қатардың, екіншісі – бағанның нөмірін білдіреді.

Екі өлшемді массивтердің типтер бөлімінде сипатталуы:

```
CONST N = 100; M = 100
TYPE Mas = Array [1..N, 1..M] Of Real;
VAR A: Mas;
```

Екі өлшемді массивтердің айнымалылар бөлімінде сипатталуы:

```
CONST N=100; M=50;
VAR A: Array [1..N, 1..M] Of Real;
B: Array[1..3,1..7] Of Integer;
```

Екі өлшемді массивтердің элементтерін ендіру немесе шығару үшін екі цикл ұйымдастырылады, сыртқы цикл-қатарлар бойынша және ішкі цикл –бағаналар номері бойынша . {N-қатарлар саны, M-бағаналар саны}.

```
Readln(N,M);
For K:=1 to N Do
For I:=1 to M do {Матрицаның элементтерін ендіру циклі}
Read(A[K,I]);
```

```
Readln(N,M);
For K:=1 to N Do
For I:=1 to M do {Матрицаның элементтерін шығару циклі}
Write(A[K,I]);
```

Бұл жерде K, I айнымалылары цикл параметрлері мен массив индекстерінің қызметтерін атқарып тұр.

1. Матрицадағы оң элементтер санын табу.

```
PROGRAM Matr;
CONST N=3; M=4;
VAR A: Array [1..N, 1..M] Of Integer;
K, I, S: Integer;
BEGIN {матрица элементтерін ендіру циклі}
For K:= 1 To N Do
For I:= 1 To M Do Read (A[K, I]);
S:=0 {оң элементтерінің қосындысын табу циклі}
For K:= 1 To N Do
For I:= 1 To M Do If A[K, I]>0 Then S:=S+1;
Writeln ('оң элементтер саны S=', S);
Readln END.
```

2. Матрицаның екінші қатарындағы теріс элементтер қосындысы.

```
PROGRAM Matr;
CONST N=10; M=7;
VAR   A: Array [1..N, 1..M] Of Real;
      K, I: integer; S: Real;
BEGIN For K:= 1 To N Do
      For I:= 1 To M Do Read (A[K, I]);
S:=0
For K:= 2 To 2 Do
For I:= 1 To M Do If A[K, I]<0 Then S:=S+A[K, I];
Writeln ('тер.эл қосындысы =', S);
Readln END.
```

3. Матрицаның соңғы екі бағанындағы тақ элементтердің қосындысын табу.

```
PROGRAM Matr;
VAR   A: Array [1..10, 1..7] Of Integer;
      K, I, S: Integer;
BEGIN For K:= 1 To 10 Do
      For I:= 1 To 7 Do Read (A[K, I]);
S:=0; For K:= 1 To 10 Do
      For I:= 5 To 7 Do
IF Odd (A[K,I]) Then S:=S+A[K,I];
Writeln ('тақ.эл қос. =', S);
Readln END.
```

4. Матрицаның ең кіші элементін табу.

```
PROGRAM Matr;
CONST N=3; M=4;
VAR   A: Array [1..N, 1..M] Of : Integer;
      K, I, Min: Integer;
BEGIN For K:= 1 To N Do For I:= 1 To M Do Read (A[K, I]);
Min:=A[1,1]; { матрицаның ең кіші элементін табу циклі }
For K:= 1 To N Do
For I:= 1 To M Do If A[K,I]<Min Then Min:=A[K,I];
Writeln ('MIN =', Min);
Readln; Readln END.
```

5. Символдық шамалардан тұратын матрицаның элементтерін алфавит бойынша реттеу.

```
PROGRAM Matr;
CONST N=3; M=4;
VAR   A: Array [1..N, 1..M] Of Char; K, I, P : Integer; Min:Char;
BEGIN
For K:= 1 To N Do
For I:= 1 To M Do Read (A[K, I]);
Writeln;
      { матрица элементтерін алфавит бойынша реттеу циклі }
For K:= 1 To N Do Begin
For P:= 1To M-1 Do
For I:= P To M Do
If A[K,P]> A[K,I] Then Begin Min:=A[K,P]; A[K,P]:=A[K,I];
```

```

A[K,I]:=Min; End, End;
    { жаңа реттелген матрица элементтерін шығару циклі }
Writeln ('Жаңа матрица: ');
For K:= 1 To N Do Begin
For I:= 1 To M Do Write (A[K,I]); Writeln End;
Readln END.

```

6. Элементтері бүтін сандардан тұратын матрица қатарларын кему бойынша реттеу және элементтері тек жұп сандардан тұратын қатарларды көрсету.

```

PROGRAM Jupkatar;
CONST N=3; M=4;
VAR A: Array [1..N, 1..M] Of Integer;
    K, I, P, Min, Jup: Integer;
BEGIN For K:= 1 To N Do
    For I:= 1 To M Do Read (A[K, I]);
Writeln; { матрица элементтерін реттеу циклі }
For K:= 1 To N Do Begin
For P:= 1 To M-1 Do
For I:= P To M Do
If A[K,P]> A[K,I] Then Begin Min:=A[K,P]; A[K,P]:=A[K,I];
A[K,I]:=Min; End, End;
Writeln ('Жаңа матрица: ');
For K:= 1 To N Do Begin Jup:=0;
For I:= 1 To M Do Begin Write (A[K,I]:4);
If A[K,I] Mod 2=0 Then Jup:=Jup+1; End;
If Jup=M Then
Write ('Жұп сандардан тұратын қатардың нөмірі: ', K); Writeln;
End;
Readln; Readln END.

```

7. Элементтері бүтін сандардан тұратын A квадрат матрицасында бас диагональдан төмен орналасқан элементтерден бір өлшемді B массивін, одан жоғары орналасқан элементтерден бір өлшемді C массивін, ал бас диагональда орналасқан элементтерден бір өлшемді D массивін құрып экранға шығару.

```

PROGRAM Jupkatar;
CONST N=4; M=N;
VAR A: Array [1..N, 1..M] Of Integer;
    K, I, P, T, S: Integer;
    B, C, D: Array [1..N*M] Of Integer;
BEGIN For K:= 1 To N Do
    For I:= 1 To M Do Read (A[K, I]);
Writeln;
For K:= 1 To N Do
For I:= 1 To M Do
If K>1 Then Begin P:=P+1; B[P]:=A[K,I] End
Else If K<1 Then Begin T:=T+1; C[T]:=A[K,I] End
Else Begin S:=S+1; D[S]:=A[K,I] End
Writeln ('Жаңа массивтер: ');
Write ('B массиві: ');
For K:= 1 To P Do Write (B[K]:4);
Writeln;

```

```

Write ('С массиві:');
For K:= 1 To T Do Write (C[K]:4);
Writeln;
Write ('Жаңа D массиві :');
For K:= 1 To S Do Write (D[K]:4);
Readln; Readln END.

```

8. Матрица элементтерінің орнын ауыстыру.

Матрицаның вертикальді осіне симметрия элементтерінің орнын ауыстыру:

```

PROGRAM Matr;
CONST N=100;
VAR A,B: Array [1..N, 1..N] Of : Integer;
    K, I, M: Integer;
BEGIN Write ('Матрица өлшемін енгіз: '); Readln(M);
For K:= 1 To M Do
For I:= 1 To M Do Read (A[K, I]);
For K:= 1 To M Do Begin
For I:= 1 To M Do Write (A[K, I]:6);
Writeln; End;
For K:= 1 To M Do
For I:= 1 To M Do B[K, I]:=A[K, M+1-I];
For K:= 1 To M Do Begin
For I:= 1 To M Do Write (B[K,I]:7);
Writeln End;
Readln; Readln; END.

```

Бақылау сұрақтары мен тапсырмалар.

1. Массивтер ұғымы, олардың программада сипатталуы.
2. Екі өлшемді массив элементтерін ендіру және шығару.
3. Екі өлшемді массив элементтерін реттеу.
4. Массивтердің типтері бөлімінде сипатталуы.
5. Массивтердің айнымалылар бөлімінде сипатталуы.
6. Элементтері алфавит бойынша реттелген 10 кісінің фамилиясынан тұратын В массиві берілген. Экранға фамилияларды алфавиттің соңынан басталатындай етіп шығару программасын құрыңыз.
7. Класта 12 ер бала және 17 қыз бала оқиды. Олардың аты, туған жылы, бойы, салмағы т.с.с. деректер берілген. Сыныптағы бойы ең ұзын ер бала және бойы ең қысқа қыз баланың аттарын экранға шығару программасын құрыңыз.
8. Класта 12 ер бала және 17 қыз бала оқиды. Олардың аты, туған жылы, бойы, салмағы т.с.с. деректер берілген. Тізімді балалардың бойы бойынша реттеп шығару программасын құрыңыз.
9. Класта 12 ер бала және 17 қыз бала оқиды. Олардың аты, туған жылы, бойы, салмағы т.с.с. деректер берілген. Қыз балалар мен ер балаларды бөлек тізіммен шығару программасын құрыңыз.
10. Матрицадағы 0-ге тең қатарларды алып тастап, қайтадан шығарыңыз.
11. Матрицаның элементтері «Сықырлы квадрат» құратындығын анықтайтын программа құрыңыз. «Сықырлы квадрат» деп барлық қатарлар, бағандар және диагональдары бойында орналасқан элементтер қосындысы бірдей болатын матрицаны айтады.
12. Матрицаның теріс сандары жоқ қатарларындағы элементтер қосындысын табыңыз.
13. Матрицаның оң сандары жоқ бағандарындағы элементтер қосындысын табыңыз.
14. Матрицаның барлық теріс элементтерін 0-мен ауыстырып шығыңыз.

№3 Зертханалық жұмыс

Тақырыбы: Жолдық қатарлар.

Мақсаты:

Турбо Паскальда қолданылатын жолдық тұрақты, жолдық айнымалы, жол туралы түсініктерін қалыптастыру. Және оларды салыстыру, реттеу жолдарын түсіндіру.

Теориялық түсініктемелер

Программаларда кездесетін әр түрлі символдардан тұратын тізбектерді жолдық қатарлар деп атайды. Өрнектерде жолдық қатарлар апострофқа алынып жазылады.

Turbo Pascal-да жолдық қатарларға компьютердің жадынан 0 байттан 255 байтқа дейін орын беріледі (жолдық қатар ұзындығына -255 символ, ұзындықтың мәніне -1 символ).

Жолдық қатарлардан тұратын идентификаторлар String типі арқылы сипатталады. Идентификатордың типінен кейін тік жақшаның ішінде осы идентификатордың ұзындығының мәні неше символдан тұруы мүмкін екені көрсетіліп кетеді. Егер бұл мін көрсетілмесе, онда жолдық қатар ұзындығына компьютердің жадынан 255 символға дейін орын беріледі.

Жолдық қатарлы идентификаторларды типтер бөлімінде (TYPE) немесе бірден айнымалылар бөлімінде (VAR) сипаттауға болады. Сонымен қатар программалауда жолдық қатарлар тұрақты шамалар ретінде де қолданылады (таңдау селекторы тұрақтысынан басқа).

Жолдық қатарлардың сипатталу мысалдары:

```
PROGRAM Katarlar;
CONST S= 'Кайсар';      {S тұрақтысы Кайсар тіркесіне тең}
TYPE Gymn= String [10];
VAR X: String [17]; {X айнымалысының ұзындығы 17 символға дейін}
    A: String [6]; {A айнымалысының ұзындығы 6 символға дейін}
    B: String;    {B айнымалысының ұзындығы 255 символға дейін}
    C: Gymn;     {C айнымалысының ұзындығы 10 символға дейін}
.....
```

1. Жолдық өрнектермен амалдар.

Операндалары жолдық қатар болып табылатын өрнектерді жолдық өрнектер деп атайды. Идентификаторларға жолдық айнымалылардың мәнін беру үшін меншіктеу операторы қолданылады. Егер меншіктеу операторы орындалғанда, символдар саны сипаттамалар бөлімінде көрсетілген шамадан асып кетсе, онда идентификатор қатардың сол жағынан бастап көрсетілген шамаға тең болатын символдарды ғана меншіктейді, ал оң жақтағы артық символдар алынып қалады.

```
...
VAR X: String [4];
BEGIN X:= 'Gumnazy'      {'azy' символдары алынып қалып, X-ке 'Gymn' тіркесі
                        ғана меншіктеледі.}

Writeln (X);
...
```

Жолдық қатарлармен біріктіру және салыстыру амалдарын орындауға болады. Біріктіру амалы бірнеше жолдық қатарды біріктіріп шығару үшін қолданылады. Біріктірілетін жолдық қатардың ұзындығы 255-тен аспауы тиіс. Мысалы:

```
.....
A:= 'Менің'; B:= 'Қазақстанымның';
Writeln (A+' '+B) {Менің Қазақстанымның – тіркесі шығады}
X:= 'Тәуелсіздігіне';
```

```

Y: ='10 жыл';
Z: ='X+' '+Y           {Z Тәуелсіздігіне 10-жыл тіркесін меншіктейді}
Writeln (Z);           {де, осы тіркесті экранға шығарады}

```

Жолдық қатарларды салыстыру амалы екі жолдық қатарды салыстыру үшін қолданылады. Қатарларды салыстыру солдан оңға қарай ең бірінші кездесетін бірдей емес символға дейін жүргізіледі. Егер қай жолдық қатардың бірдей емес символының информация алмастыру стандартты таблицасындағы нөмірі үлкен болса, сол жолдық қатар үлкен деп есептеледі. Егер жолдық қатарлардың ұзындығы мен барлық символдары сәйкес келсе, олар тең деп есептеледі. Ал егер жолдық қатарлардың ұзындықтары әр түрлі болып, ал символдары сәйкес болып келсе, онда үлкені болып ұзындығы үлкен жолдық қатар есептеледі. Жолдық қатарлардың салыстыру амалдарының нәтижесі әрқашан бульдік тип болады.

```

'Intel' > INTEL           {True}
'Pentium' < 'PENTIUM'    {False}
'Duron'='Duron'          //True
'Celeron'<>'Celeron'     {False}
'Hewlett'<='Hewlett Packard' {True}
'Laser'>='LaserJet'      {False}

```

2. Жолдық процедуралар мен функциялар.

Жолдық Қатардың ұзындығы есептеу үшін LENGTH функциясын қолданылады, нәтиженің типі 0-ден 255-ке дейінгі бүтін сан (Byte) болып табылады:

```

X:= LENGTN (A) ; // X идентификаторы A жолдық қатарының
                 ұзындығын меншіктейді.
Writeln (LENGTN (A)) // экранға A жолдық қатарының ұзындығы шығады.

```

Төмендегі программада X жолдық қатарының ұзындығы есептеліп табылып, ол бүтін сан болатын A идентификаторына меншіктелген:

```

PROGRAM Kat1;
VAR X: String;A:integer;
BEGIN X: ='Pentium -III';
A:=Length (x);
Writeln ('A=',A); Readln END.

```

Жолдық қатардан фрагмент көшіріп алу үшін COPY функциясы қолданылады:

```

COPY (A, N, M); // A жолдық қатарының N позициясындағы символдан бастап M
                символ көшіріп алады.
X:=COPY ('Облыстық гимназия',10,4) ); // X идентификаторы "гимн"
                                       фрагмент меншіктейді.
Writeln (COPY (' Облыстық гимназия', 10,8)); // экранға гимназия сөзі
                                             шығады.

```

Төмендегі программада X қатарынан Pentium сөзі көшіріліп, A идентификаторымен белгіленген жолдық қатарға меншіктеледі.

```

PROGRAM Kat2;
VAR X, A: String;
BEGIN X: = 'Pentium-III'; A: = Copy (X,1, 7);
Writeln ('A=',A); Readln END.

```

Жолдық қатарлар фрагменттерін біріктіру үшін CONCAT функциясы қолданылады.

```

CONCAT (X,A); // функция X және A қатарларын біріктіреді.
Writeln (CONCAT(' Гимн', 'Азия')); // экранға ГимнАзия сөзі шығады.
Y:=Concat (' Гимн ', Азия '); // Y ГимнАзия сөзін меншіктейді.

```

Төмендегі программада жолдық қатар болып табылатын Y тұрақтысы X қатарымен біріктіріліп, нәтиже A идентификаторымен белгіленген жолдық қатарға меншіктеледі:

```
PROGRAM Kat3;  
CONST Y= 'Input '; VAR X,A: String;  
BEGIN X:= 'Output'; A:= Concat (Y, X);  
Writeln ('A= ',A); Readln END.
```

Жолдық қатардың ішіндегі тұрған бір фрагменттің орнын анықтау үшін POS функциясы қолданылады. Бұл функция мәнінің типі бүтін және ол жолдық қатарда берілген фрагменттің басталатын позиция нөміріне тең болады. Ал егер жолда ондай фрагмент кездеспесе онда нәтиже 0-ге тең болады, мысалдар:

```
POS (X, A); // X фрагментінің A қатарындағы басталатын нөмірі  
POS('наз', 'Гимназия'); //нәтиже 4-ке тең болады.  
POS('зал', 'Гимназия'); //нәтиже 0-ге тең болады.
```

Төмендегі программада X фрагменті Y қатарындағы басталатын орны бүтін сан болатын A идентификаторына меншіктелген.

```
PROGRAM Kat4;  
CONST Y= 'Input ';  
VAR X: String; A: Byte;  
BEGIN X:= 'put';  
A:= Pos (X, Y); Writeln ('A=', A); Readln END.
```

Жолдық қатардан бір фрагмент өшіріп тастау үшін DELETE процедурасы қолданылады, жазылуы:

```
DELETE(A, N, M); //A жолдық қатарының N позициясындағы символдан  
бастап M символ өшіріп тастайды.  
DELETE ('Облыстық гимназия', 10,4); // »Облыстық азия » тіркесі қалады.
```

Төмендегі программада X қатарының N-ші позициясынан бастап M символ өшіріледі:

```
PROGRAM Kat5;  
VAR X: String; N, M: Byte;  
BEGIN Readln (N, M); X:= 'Output';  
Delete (X, N, M); WriteLn ('X=', X); Readln END.
```

Жолдық қатарға бір фрагмент қосу үшін INSERT процедурасы қолданылады.

```
INSERT( A, X, N ); //X жолдық қатарының N-ші позициясынан бастап  
A фрагментін қою.
```

Төмендегі программада X қатарының 3 позициясынан бастап A қатары қойылып X қатары "Internet" мәніне өзгереді:

```
PROGRAM Kat6;  
VAR X, A: String; N: Byte;  
BEGIN N:=3; X:= 'Innet'; A:= 'ter'  
Insert (A, X, N, ); Writeln ('X=', X); Readln END.
```

Латын алфавитінің кіші әріптерін бас әріпке айналдыру үшін UPCASE функциясы қолданылады.

```
UPCASE(X); ); //X жолдық қатарындағы кіші әріптерді бас әріптерге  
өзгертеді.
```

Төмендегі программаларда X символы («a») мен X қатарындағы («ә») кіші әріптер бас әріптерге айналдырылған:

```
PROGRAM Kat7;
```

```
VAR X: Char; K: Byte;
BEGIN X:= 't'; X:=UPCASE(X)
Writeln ('X=',X); Readln END.
```

ә) PROGRAM Kat8;

```
VAR X: String; K:Byte;
BEGIN X:='Internet'; For K:=1 To Length (X) Do X[K]:= UPCASE(X[K]);
Writeln ('X=',X); Readln END.
```

STR процедурасы кез – келген сандық шаманы бір жолдық қатарға қою үшін қолданылады, жазылуы:

```
STR (150, X); // 150 саны X жолдық қатарының ішіне орналасады.
```

Төмендегі программада A нақты санын X жолдық қатарына айналдырып, әрі қарай X қатары Y қатарын шығаруда қолданылған:

PROGRAM Kat9;

```
VAR X, Y, S: String; A: Integer;
BEGIN A:=134; S:='TU'; Str (A,X); Y:=S+X;
Writeln ('Y=',Y); Readln END.
```

Жолдық қатарды сандық шамаға айналдыру үшін VAL процедурасы қолданылады, жазылуы:

```
VAL ( X, A, C);
```

Төмендегі программада X жолдық қатарын A нақты санына айналдырып, әрі қарай A шамасы Y өрнегін есептеп табуда қолданылған:

PROGRAM Kat10;

```
VAR X: String; Y, A: Real; C: Integer;
BEGIN X:= '2.5E+4';
Val (X, A, C); Y:=A/2-5*A;
Writeln ('A=', A: 3:2); Writeln ('Y=', Y:3 :2); Readln END.
```

3. Жолдық қатарларға программа құру.

1. Әр түрлі тіркестермен амалдар орындау.

а)

PROGRAM Katar1;

```
VAR X: String;[17]; Y, A, B, C: String; N, M, Z, P: Integer;
```

```
BEGIN X:= 'Облыстық гимназия';
```

```
Z:= Length (X); {ұзындығын анықтау}
```

```
P:= Pos ('азия', 'Облыстық гимназия');
```

```
Readln (N, M);
```

```
Y:= Copy (X, N, M); {жолдан фрагмент ойып алу}
```

```
A:= Copy (X, 1, 3); B:=(X,10,8); C:=Concat (A, B);
```

```
{X тізбегінен Облгимназия сөзін шығару}
```

```
Writeln (Z, ' ', Y, C) Readln END
```

ә)

PROGRAM Katar2;

```
CONST X = 'Облыстық'; Y= 'гимназия';
```

```
VAR A, B, C, D: String[15]; K: String; I, N: Byte;
```

```
BEGIN A:= 'Pentium-I'; B:= 'Pentium-II'; C:= 'Pentium-III';
```

```
If A > B Then D:=A Else D:=B; {3 қатарды салыстырып үлкенін табу}
```

```
If D > C Then Writeln ('max=', D) Else Writeln ('max =', C);
```

```

K:=X+' '+Y; {қатарларды біріктіру}
If K=Concat (X, Y) Then Writeln ('K=X+Y') Else Begin
Writeln ('K ten emes X+Y'); Writeln ('k=', K) End; {2 қатарды салыстыру}
Delete (A, 8, 2,); Writeln ('A=', A); {қатардан фрагмент алып тастау}
Delete (B, 2, 6,); Writeln ('B=', B); {қатардан фрагмент алып тастау}
Insert ('entium', B,2); Writeln ('B1=',B); {қатарға фрагмент қою}
K:=Copy(Y,1,4); {қатардан фрагмент көшіріп алады}
Writeln ('K1=',K);
I:=Pos(K,Y); Writeln ('I=',I); {Y қатарында K фрагментінің орнын анықтау}
N:=Length(X+Y); Writeln ('N=',N); {қатар ұзындығын анықтау}
Writeln (Concat (x', ',y)); {қатарларды біріктіріп шығару}
Writeln ('a-max=', Uprcase('a')); {бас әріп шығару}
Readln END

```

2. Кез – келген жолдық қатардың латын алфавитінің кіші әріптерін бас әріпке айналдыру программасын құру.

```

PROGRAM Kat3;
VAR X: String; K: Byte;
BEGIN Writeln('Қатардың мәнін енгіз:'); Readln(X);
For K:=1 to Length(x) do
X[K]:= Uprcase(X[K]);
Writeln ('Қатардың жаңа мәніX=',X); Readln END.

```

3. X жолдық қатарында кездесетін 'а' символдарын санап, оларды 'b' символымен ауыстыру программасын құру.

```

PROGRAM Kat4;
VAR X: String; K, N: Byte;
BEGIN Write ('қатардың мәнін енгіз'); Readln (X);
For k:=1 to Length(X) do X[K]='a' then
Begin n:=n+1; X[K]='b' end;
Writeln ('қатардың жаңа мәні X= ', X, ' қатарда ', N, ' "a" символы болады'); Readln END.

```

4. X қатарындағы фигуралық жақшалар ішінде тұрған фрагменттерді алып тастап шығару программасы.

```

PROGRAM Katar 6;
Type Katar= String;
VAR X: Katar; K, I: Byte;
Begin Writeln('Қатардың мәнін енгіз:');Readln(X);
Repeat
K:=Pos('{',X); If K>0 Then Begin I:=Pos('}',X);
if I>K Then Delete (X, K, I-K+1) End;
Until K=0;
Writeln ('қатардың жаңа мәні X=',X); Readln END.

```

5. Берілген сөздің екі жаққа да бірдей жазылып оқылатынын анықтау программасын құру.

(мысалға: қызық,қырық , қазақ, ата, ана, ара, нан, шаш, т.с.с сөздер)

а)

```
PROGRAM Kat 7;
VAR X,Y: String; K, N: Byte;
BEGIN Write ('қатардың мәнін енгіз:'); Readln (X);N:=0;
For K:=Length(X) Downto 1Do Y:=Y+Copy (X,K,1);
If X=Y Then Writeln ('екі жаққа да бірдей') Else Writeln ('бірдей емес');
Readln END.
```

ә)

```
PROGRAM Kat 7;
VAR X: String; K, N: Byte;
BEGIN Write ('қатардың мәнін енгіз'); Readln (X);N:=0;
For K:= 1 To Length(X) Div 2 Do
If X[K]=X[(Length(X) +1-K) Then N:=N+1;
If N=(Length(X)Div 2) Then Writeln ('қатар екі жаққа да бірдей оқылады');
Else Writeln ('оқылмайды');
Readln END.
```

6. Берілген футбол командалары тізімінен Т әрпінен басталатын футбол командаларын алып тастап, қайта шығару программасы.

```
PROGRAM Katar 8;
CONST T: Array [1..5] of String= ('Tobol', 'Kaisar', 'Irtys', 'Taraz', 'Tor');
VAR K,N, I : Byte;
BEGIN For K:= 1To 5 Do Write (T[K] :7); Writeln;
N:=5; K:=0; Repeat K:=K+1;
Write Copy (T [K], 1 , 1)= 'T' Do Begin I:=K;
Repeat T [I]:= T [I+1]; I : =I+1; Until I>=N;
T[N]:=''; N:=N-1; End;
Until K>=N;
Writeln ('жаңа тізім:');
For K:=1 To N Do Write (T[K]:7);
Readln END.
```

7. Берілген футбол командалары аттарын кері шығару программасы (мысалға: Tobol сөзі lobo T болып шығу керек).

а) 1түрі

```
PROGRAM Katar 9;
CONST T: Array [1..5] of String= ('Tobol', 'Kaisar', 'Irtys', 'Taraz', 'Kairat');
VAR K,I : Byte; S: Char; P: Array [1..5] Of String;
BEGIN For K:= 1To 5 Do Write (T[K] :7); Writeln;
For K:= 1To 5 Do Begin P[K]:='';
For I:= Length (T[K]) Downto 1Do P[K]:=P[K]+Copy (T[K],I,1) end;
Writeln ('жаңа массив :');
For K:= 1To 5 Do Write (P[K] :7); Readln END.
```

ә) 2түрі

```
PROGRAM Katar 9;
CONST T: Array [1.. 5] of String= ('Tobol', 'Kaisar', 'Irtys', 'Taraz', 'Kairat');
VAR K,I : Byte; S: Char;
BEGIN For K:= 1To 5 Do Write (T[K] :7); Writeln;
For K:= 1To 5 Do
```

```

For I:=1 To Trunc (Length(T[K])/Do
Begin S:=T[K][I]; T[K][I]:= T[K][ Length(T[K])-I+1];
T[K][ Length(T[K])-I+1]:=S End;
Writeln ('жаңа массив футбол командларының кері аттарынан тұрады:');
For K:= 1To 5 Do Write (T[K] :7); Readln END.

```

Бақылау сұрақтары мен тапсырмалар.

1. Жолдық қатарлар дегеніміз не?
2. Жолдық қатарлардың сипатталуы.
3. Жолдық қатарларға қандай амалдар қолдануға болады?
4. Жолдық қатардың ұзындығы қандай?
5. Жолдық қатардан фрагмент қалай көшіруге болады?
6. Жолдық қатарларды біріктіру үшін қандай функция қолданылады?
7. Жолдық қатардың ішінде орналасқан фрагменттің орны қалай анықталады?
8. Жолдық қатардың ішінен фрагмент өшіру үшін қандай процедура қолданылады?
9. Жолдық қатарға фрагмент қосу үшін қандай процедура қолданылады?
10. Латын алфавитінің кіші әріптерін бас әріп етіп шығару үшін қандай функция қолданылады?
11. Сандық шаманы жолдық қатарға айналдыру үшін қандай процедура қолданылады?
12. Жолдық қатарды сандық шамаға айналдыру үшін қандай процедура қолданылады?
13. Клавиатурадан енгізілген қатардағы барлық әріптер санын табу программасын құрыңыз?
14. Клавиатурадан енгізілген қатар символдары сандық шамалардан тұратындығын анықтайтын программа құрыңыз?
15. Клавиатурадан енгізілген қатар символдарын өсу бойымен реттеп шығаратын программа құрыңыз?
16. Клавиатурадан енгізілген қатардағы ең ұзын сөзді табыңыз?
17. Клавиатурадан енгізілген қатардағы ең көп кездесетін әріпті табу программасын құрыңыз?
18. Клавиатурадан енгізілген қатардағы бірінші символмен сәйкес келетін барлық символдарды алып тастау программасын құрыңыз?
19. А қатары берілген. Қатардағы қайталанбайтын символдар санын табу программасын құрыңыз?
20. А қатары берілген. Қатардың жұп нөмірлі орындарында тұрған символдарын өшіріп тастау программасын құрыңыз?
21. Екі А және В жолдық қатарлары берілген.
22. Екі А және В жолдық қатарлары берілген. А қатарында В қатары неше рет кездесетін анықтау программасын құрыңыз?
23. Екі А және В жолдық қатарлары берілген. А қатарындағы символдардан В қатарын құрастыруға болатынын анықтау программасын құрыңыз?
24. Екі А және В жолдық қатарлары берілген, А қатарымен В қатарында бір орында тұрған бірдей символдардың санын табу программасын құрыңыз.
25. Екі А және В жолдық қатарлары берілген, А қатарында да В қатарында да кездесетін барлық символдардың санын табу программасын құрыңыз.
26. Екі А және В жолдық қатарлары берілген, А қатарында да В қатарында да кездесетін барлық символдарды алып тастау программасын құрыңыз.
27. Екі А және В жолдық қатарлары берілген, А қатарында да В қатарында да кездесетін барлық символдардан тұратын С жолдық қатарын шығару программасын құрыңыз.
28. Екі А және В жолдық қатарлары берілген, А қатарында да В қатарында да кездесетін барлық жұп нөмірлі орындарда тұрған символдарды анықтаңыз.

№4 Зертханалық жұмыс

Тақырыбы: :Жиындар.

Мақсаты:

Турбо Паскальда қолданылатын жиындар .Жиындарды сипаттау туралы түсініктерін қалыптастыру. Жиын элементтерімен қандай амалдар орындауға болатындығын түсіндіру.

Теориялық түсініктемелер

Математикада жиындар деп белгілі бір қасиеттері жағынан бір-бірімен жақын байланысқан элементтердің жиынтығын айтады. Мысалы түзу, тік төртбұрыш, шеңбер, ромб, квадраттар жазықтықтағы фигуралар жиыны, ал куб, конус, пирамида, призмалар кеңістіктегі фигуралар жиыны болып табылады. Turbo Pascal-да жиындар деп бір типті элементтердің жиынтығын айтады. Программда жиындар белгілі бір идентификатормен белгіленеді. Жиындарға кіретін элементтер типін негізгі тип деп атайды. Негізгі тип ретінде нақты типтерден басқа барлық скалярлық типтердің бәрін қолдануға болады. Программда жиындар айнымалылар (VAR) немесе типтер (TYPE) бөлімінде сипатталады. Жиындар типін сипаттау үшін SET OF тіркесі қолданылады. Жиынды құрайтын элементтер санын жиынның қуаты деп атайды, оның оның ұзындығы 256-дан аспауы тиіс. Жиындардың айнымалылар бөлімінде жазылуы мен сипатталуы:

VAR жиын аты: Set of негізгі тип;

VAR A: Set Of 1..31;

B: Set Of Char;

Мұндағы: A,B – жиын аттарын білдіретін идентификатор,

Set – жиынды білдіретін қызметші сөз.

Жиындардың типтер бөлімінде жазылуы мен сипатталуы:

TYPE тип аты = Set Of негізгі тип;

VAR жиын аты: тип аты;

TYPE Jy=Set Of 1..31;

VAR A: Jy;

Мұндағы: Jy- жиындағы элементтердің ортақ типі,

1..31 – жиын элементтерінің негізгі типі, 1-ден 31-ге дейінгі бүтін сандар,

A- жиын аты ретінде белгіленген идентификатор.

Өрнектерде жиын элементтерінің мәндері арнаулы конструктор арқылы беріледі. Конструктор деп тік жақшалардың ішінде бір-бірінен үтір арқылы жазылатын негізгі типті элементтер тізімін айтады, егер жиынның құрамында ешқандай элемент жоқ болса, онда ондай жиынды бос жиын деп атайды да, тік жақшаның іші бос болып жазылады. Жиындардың өрнекте жазылу мысалдары:

A:=[1,5,2,4,3]; //A жиыны 5 бүтін сандардан тұрады

B:='A', 'B', 'C', 'D']; //B жиыны 4 символдан тұрады

C:=[] //C жиыны бос жиын

Жиын элементтерімен келесі амалдар орындауға болады:

1. **X:=A+B** (жиындарды біріктіру) – Екі жиынның бірігу нәтижесінде екі жиынның барлық элементтерінен тұратын үшінші жиын шығады.
2. **X:=A*B** (жиындардың қиылыстыру) –екі жиынның қиылысу нәтижесінде тек екі жиында да бар элементтерден тұратын үшінші жиын шығады.
3. **X:=A-B** (бір жиыннан екінші жиынды алу) – Бір жиыннан екінші жиында алып тастау нәтижесінде бірінші жиынның екінші жиында жоқ элементтерінен тұратын үшінші жиын шығады.

Бұлармен қатар жиындарға төмендегідей логикалық салыстыру амалдарын қолдануға болады:

1. $A=B, A \supset B$ – Егер А жиынының барлық элементтері В жиынында бар болса, ал В жиынының барлық элементтері А жиынында бар болса және олардағы элементтер саны бірдей болса, онда А мен В жиындары тең, әйтпесе бұл жиындар тең емес деп есептеледі.
2. $A \subseteq B, B \supseteq A$ - Егер А жиынының барлық элементтері В жиынының элементі болып табылса, онда А жиыны В жиынына тиісті болып есептеледі.
3. $X \in B$ - X шамасының В жиынының элементі болып табылатындығын анықтау функциясы.

Сондай-ақ **In** функциясын құрама шарттардың орнына қолдануға болады, мысалға:

While (K>=1) And (K<=6) орнына **While k In [1..6] Do** деп жазуға, немесе: **If (X=1) Or (X=2) or (X=3) Or(X=4) Or(X=5)** орнына **If k In [1..5]** деп жазуға болады.

Программалауда жиындардың қолданылуының көптеген артықшылықтары бар: жиындар компьютер жадынан өте аз орын алып, программаның жадтан алатын орны үнемделеді (жиынның 1 элементі 1 бит орын алады); тармақталу операторларының жазылуын жеңілдетеді, программаның компиляциялану мен орындалу уақыты азаяды. Сонымен бірге жиындардың программалауда кездесетін басты кемшілігі: оларды енгізу-шығару мүмкіндіктерінің жоқтығы болып табылады (ол үшін программалаушы арнайы процедуралар жазып, программаның жазылу процесі біршама уақытқа созылып кетеді).

4.1. Жиындарға программа құру.

1. A, B, C жиындары берілген, $X=(A*B) + (A-B)$ болатын жаңа жиынды экранға шығарып, C жиынының барлық элементтерінің X жиынының элементтері болып табылатынын анықтау.

```
PROGRAM J1yn 1;
VAR A, B, C, X: Set Of Char; Y: Char;
BEGIN A:= ['A', 'B', 'D', 'R', 'T', 'X', 'Y', 'Z'];
B:= ['A', 'D', 'T', 'N', 'U', 'X', 'Z'];
C:= ['A', 'D', 'X', 'R', 'Z']; X:=(A*b)+(A-B);
Write ('X=[ '); For Y:= 'A' To 'Z' Do
If Y In X Then Write ( ' ', Y); Write ( ' ') : Writeln;
If C <=X Then Write ( ' C жиыны X жиынына тиісті')
Else Write ( ' Тиісті емес') ; Readln END.
```

2. 1-ден 25-ке дейінгі бүтін сандардан тұратын А жиынына тиісті Y жиынын жасау программасы.

```
PROGRAM J1yn 2;
VAR A, Y: Set Of 1..25; X, K: Byte;
BEGIN For K:=1 To 25 Do A:=A+[K];
For K:=1 To 10 Do Begin Writeln ( ' Сандар енгіз'); Readln (X);
If (X In A) Then Begin Writeln ( ' Бұл сан А жиынына тиісті') ; Y:=Y+[X]; end Else
Writeln(' А жиынында бұндай сан жоқ ' ) ; End;
Writeln(' Жаңа Y жиыны мынадай сандардан тұрады:'); For K:=1 to 25 Do If K In Y Then
Write ( ' Y=', K:3);
Readln
END.
```

3. 1-ден 50-ге дейінгі бүтін сандар ішінен 2,3,4,5 сандарына 3,4 сандарына және 4-ке де 5-ке де қалдықсыз бөлінетін сандар шығару.

```
PROGRAM Jiyn3;
CONST N=50;
TYPE But=Set Of 1..N;
VAR X2,X3,X4,X5,X34,X45: But; K: Byte;
BEGIN X2:[]; X3:[]; X5:=[];
For K:=1 To N Do Begin
If K Mod 2=0 Then X2:=X2+[K];
If K Mod 3=0 Then X3:=X3+[K];
If K Mod 4=0 Then X4:=X4+[K];
If K Mod 5=0 Then X5:=X5+[K]; End;
X34:=X3+X4; X45:=X4*X5;
Write ('X2='); For K:=1 To N Do If K In X2 Then Write (K:3) ; Writeln;
Write ('X3='); For K:=1 To N Do If K In X3 Then Write (K:3) ; Writeln;
Write ('X4='); For K:=1 To N Do If K In X4 Then Write (K:3) ; Writeln;
Write ('X5='); For K:=1 To N Do If K In X5 Then Write (K:3) ; Writeln;
Write ('X34='); For K:=1 To N Do If K In X34 Then Write (K:3) ; Writeln;
Write ('X45='); For K:=1 To N Do If K In X45 Then Write (K:3) ;
Readln END.
```

4. берілген сөйлемдегі дауысты және дауыссыз дыбыстар санын анықтау.

```
PROGRAM Jiyn 4;
VAR A: Set Of Char; X: String; N,K,M: Byte;
BEGIN
A:=['A','a','Ә','ә','E','e','И','и','I','i','O','o','Ы','ы','У','у','Ү','ү','Ү','ү','Э','э','Ю','ю','Я','я'];
Write ( 'Сөйлем жазып енгіз. '); Readln (X) ; N:=0; M:=0;
For K:=1 TO Length(X) Do Begin
If X [k] In A Then N:=N+1; {дауысты дыбыстар санын анықтау}
If X[K]=' ' Then M:=M+1 END; {бос орындар санын анықтау}
Writeln (N, 'дауысты және ' Length(X) -N-M, 'дауыссыз дыбыстар бар');
Readln END.
```

5. берілген сөйлемнен кез келген сөз алып тастау

(Сөздің фрагменттен айырмасы екі жағында бос орын болады)

```
PROGRAM Jiyn 5;
VAR K,L: Integer; A,B: String; M: Set Of Char;
BEGIN M:=['.', ',', '!', ':', ';', '?', '-', ' ']; {тыныс белгілері}
Readln (A); A:= Concat ( ' ',A);
Readln(B); B:= ' '+B;
L:= Length (A);
Repeat
K:= Pos (A,B);
```

If (K<>0) And (B[K+L] IN M Then Delete (B,K,L); {сөзді алып тастау}
Until K=0;
Writeln (B); Readln END.

Бақылау сұрақтары мен тапсырмалар.

1. Жиындар дегеніміз не?
2. Жиындардың сипатталуы.
3. Жиындардың негізгі типі дегеніміз не?
4. Конструктор дегеніміз не?
5. Бос жиындар дегеніміз не?
6. Жиын элементтерімен қандай амалдар орындауға болады?
7. Жиындар қандай жағдайда тең жиындар деп есептеледі?
8. Қандай жағдайларда бір жиын екінші жиынға тиісті болып есептеледі?
9. Жиындардың бірігуі дегеніміз не?
10. Жиындардың қиылысуы дегеніміз не?
11. Бір жиыннан екінші жиынды алып тастау нәтижесі не береді?
12. IN функциясы не үшін қызмет атқарады?
13. Программалауда жиындардың қолданылуының артықшылықтары мен кемшіліктерін тізіп беріңіз.
14. A[1..30] жиынын сипаттаңыз.
15. A['a'..'b'] жиынын сипаттаңыз.
16. A[1..20] жиынына екі таңбалы палиндром сандарды орналастыру программасын құрыңыз.
17. A[1,2,3,4,5] және B[1,3,5] жиындарын салыстыру программасын құрыңыз.
18. A['a', 'b', 'c', 'd'] B['b', 'c', 'a', 'd'] тең екенін анықтау программасын құрыңыз.
19. A[2,4,6,8,10] және B[1,3,5,7,9] жиындары берілген. B жиынының A жиынына тиісті емес екенін анықтаңыз.
20. A[10,20,30,40,50] және B[10,30,50] жиындары берілген. B жиынының A жиынына тиісті екенін анықтау программасын құрыңыз.
21. A[1,5,7,18,110] және B[2,13,145,177,259] жиындары берілген. Екі жиынды біріктіру программасын құрыңыз.
22. A[5,10,15,20,25,30,35,40,45,50] және B[10,30,50,60,70,80,90,100] жиындары берілген. Екі жиынды қиылыстыру программасын құрыңыз.

№5 Зертханалық жұмыс

Тақырыбы: Жазулар

Мақсаты:

Турбо Паскальда қолданылатын жазулар .Жазуларды сипаттау туралы түсініктерін қалыптастыру. Жазулар программалық тексте қалай жазуға болатындығын түсіндіру.

Теориялық түсініктемелер

Ғылыми –техникалық және экономикалық есептеулерде көптеген мәліметтерді өңдеуде оладың типтерінің әртүрлі болуы біршама қиындықтар туғызады. Мысалы көп өлшемді массивтердің элементтерінің типтерін бір түрге келтіру үшін бірқатар қосымша функциялар (Val,Str) қолдануға тура келеді. Кейбір есептеулерде әр түрлі типті мәліметтердің әрқайсысына бөлек массив қолданылады да, бір –бірімен индекстері арқылы сәйкестендіріледі. Бұлар программалық тексті үлкейтіп және оның оқылуын ауырлатып жібереді. Осындай жағдайларда программаның құрылымын анағұрлым жеңілдету үшін әр түрлі типті мәліметтер жиынтығы болып табылатын жазуларды (немесе жазбалар) қолданылады.

Программада жазулар белгілі бір идентификатормен белгіленіп, типтер бөлімінде сипатталады. Типтер бөлімінде жазуларды сипаттау жазуларды білдіретін RECORD(жазу,жазба) сөзімен басталады да, END қызметші сөзімен аяқталады. Олардың аралығында жазулар өрісі деп аталатын жазулар құрамына кіретін барлық компоненттердің тізімі олардың типтерімен бірге көрсетіледі. Жазулардың типтер бөлімінде сипатталу мысалдары:

a) TYPE Dosye = RECORD

A,C: String;

B,E: Integer;

D:Char; END;

VAR X,Y: Dosye;

мұндағы:

-Dosye -барлық жазу компоненттерінің ортақ типі ретінде белгіленген идентификатор;

- RECORDS... END жазуды анықтайтын операторлық жақшалардың қызметін атқарады;

-жақшалардың ішінде жазулар өрісі идентификаторлары сипатталған;

A,C-жолдық қатар болатын айнымалылар (мыс: фамилия, аты,);

B,E- бүтін сандар (мыс: туған жылы, оқитын класы),

D- символдық шама (мыс. қан тобы),

X,Y- жазу аты ретінде белгіленген жазулық айнымалылар (X және Y идентификаторлары

Dosye- типті жазуларды анықтайды).

ә)

TYPE Futbool= RECORD

A:Array [1..17] Of String;

B,C: Array [1..17] Of Byte;

D:Array [1..17] Of Integer; END;

VAR X: Futbool;

Мұндағы:

-Futbool - барлық жазу компоненттерінің ортақ типі ретінде белгіленген идентификатор;

- RECORDS... END жазуды анықтайтын қызметші сөздер операторлық жақшалардың қызметін атқарады;

-операторлық жақшалардың ішінде жазулар өрісінің айнымалылары сипатталған;

A- элементтері жолдық қатардан тұратын массив (мыс: командалар аты);

B, C, D- бүтін сандар массивтері (мыс: жеңістерінің, тең өткізген ойындар, ұпайлар сандары);
X, Y – жазу аты ретінде берілген жазулық айнымалылар (X және Y идентификаторлары
Football –типті жазуларды анықтайды).

Көптеген есептеулерде жазуды анықтайтын идентификатор ретінде элементері жазу өрістері болатын массивтерді қолданған ыңғайлы болып табылады, олар программада төмендегідей сипатталады:

TYPE Dosye= RECORD

A,C: String;

B,E:Integer;

D:Char;

END;

VAR X: Array [1..10] Of Dosye;

мұндағы:

-Dosye – жазулардың аты ретінде белгіленген идентификатор;

- RECORDS... END жазуды анықтайтын операторлық жақшалардың қызметін атқарады;

- Жақшалардың ішінде жазулар өрісі идентификаторларды сипатталған;

A,C-жолдық қатар болатын айнымалылар (мыс: фамилия, аты,);

B,E- бүтін сандар (мыс: туған жылы, оқитын класы),

D- символдық шама (мыс. қан тобы),

X - жазулық массив (X массиві Dosye-типті жазуларды анықтайды).

Жазу өрістерінің мәндерін өрнектегенде қолдану үшін айнымалылар құрама атпен жазылады. Айнымалылардың құрама аты: жазуды анықтайтын идентификатордан, нүктеден және жазу өрісінің атынан тұратын тізбектен тұрады. (X. A, X. C, X. E). Құрама атпен белгіленген жазу өрістерінің айнымалылары өздерінің көрсетілген типтерімен орындауға болатын барлық есептеулерге қатыса алады, мысалға жазулармен төмендегідей амалдар орындалады:

Y = X // Y жазуы X жазуының барлық компоненттерін меншіктейді.

X. A:= 'Ахметова'; // A айнымалысы Ахметова қатарын меншіктейді

X. C:1987; //C айнымалысы 1987 санын меншіктейді

X.D.:='B'; //D айнымалысы B символын меншіктейді

Read(X.E); //E айнымалысының мәнін енгізу

Writeln(' Фамилия: ',X.A); //A айнымалысының мәнін шығару

X.B<X.E; // Логикалық салыстыруамалдарын қолдану

X.B>=1985;

X.B<=1988;

X.E<>11;

X.A='Мергенов Самат';

Программалауда жазу өрісінің айнымалыларын құрама атпен жазу бірқатар қиындықтар туғызады(программалық текст ұзақ болып кетеді және оны теру программистің көп уақытын алады). Сондықтан Turbo Pascal-да жазу айнымалыларын біріктіру үшін WITH операторын қолданылады. Жазулар айнымалыларының алдында WITH операторын жазып кетсе, онда олардың алдында жазуды анықтайтын идентификаторды жазбаса да болады, жалпы жазылу түрі:

With X Do операторлар бөлімі;

Мысалы:

With X Do Readln (A);

Жазуды анықтайтын идентификатор массив болған жағдайда жанына тік жақшалар ішіне массивтің индексі қосыла жазылады:

With X[K] Do Readln (A);

Егер жазу айнымалыларынан тұратын операторлар бөлімі бірнеше операторлардан тұрса, олар операторлық жақшаға алынады:

With X Do Begin операторлар бөлімі End;

Мысалы:

```
For K:= 1 To 3 Do
With X[K] Do Begin
Write('Фамилиясы: '); Readln(A); Write('Аты: '); Readln(C);
Write('Туған жылы: '); Readln(B); Write('Класы: '); Readln(E);
Write('Тобы: '); Readln(D); End;
```

Егер жазулар бірінің ішінде бірі орналасса, онда With операторы да бірінің ішіне бірі орналасады. Олардың жазылу принципі цикл ішінде цикл ұйымдастыру принципіне ұқсас келеді.

```
With X Do Begin
With X Do Begin
Write('Фамилиясы: '); Realdln(A);
Write('Аты: ') Realdln(C); End;
Write('Туған жылы: '); ); Realdln(B);
Write('Класы: '); Realdln(E); End;
```

5.1. Жазуларға программалар құру.

1. Жүргізушілерге картотека жазу программасы.

```
PROGRAM Dosye;
USES Crt;
TYPE Dos=RECORD
    A:Char;
    B:Integer;
    C: String;
    D: String;
    E: String; End;
VAR X: Array[1..10] Of Dos;
    B,K,I: Integer;
Procedure Inp;
Begin Writeln(K, ' машина туралы деректерді енгізіңіз');
```

```
    With X[K] Do Begin
        Write('Қаласы: ');
        Readln(A);
        Write ( ' машина нөмірі: ');
        Realdln(B);
        Write('Сериясы: ');
        Readln(C);
        Write('Фамилиясы,аты: ');
        Realdln(D);
        Write('Адрес: ');
        Readln(E); End; End;
```

```
Procedure Out ( I: Integer);
Begin Writeln(, ' Бұл нөмірмен мынадай машиналар бар: ');
With X[K] Do Begin
```

```

Writeln ('Қаласы:',A);
Writeln ( ' Машина нөмірі:', B);
Writeln('Сериясы:', C);
Writeln('Фамилиясы,аты: ',D);
Writeln('Адрес:',E); End; End;

```

```

BEGIN Clrscr;
For k:=1 To 3 Do Inp;
Write ('Керекті машина нөмірін енгіз: '); ReadLn(B);
For K:= 1 To 3 Do
If X[K]. B=B Then Out (k);
ReadLn End.

```

2. Оқушыларға картотека жазу программасы.

а) 1 түрі:

```

PROGRAM Jazy 1;
TYPE Dosye= RECORD
A, C: String;
B, E :Integer;
D: Char;
End.
VAR X: Dosye;
Begin
Write('Фамилиясы:'); Realdn(X.A);

Write('Аты:'); Readln(X.C);
Write('Туған жылы :'); Readln(X.B);
Write('Класы:'); Readn(X.E);
Write ('қан тобы:'); Readln(X.D);
Writeln('X.A,' ', X.C,' ', X.B, ' ', X.E, ' ', X.D, ' ');
Readln END.

```

ә) 2 түрі:

```

PROGRAM Jazy 2;
TYPE Dosye= RECORD
A: String;
B :Integer;
C: String;
D: Char;
E: Integer;
End.

VAR X: Dosye;
Begin
With X do begin
Write('Фамилиясы:'); Readln(A);
Write('Аты:'); Readln(C);
Write('Туған жылы :'); Readln(B);
Write('Класы:'); Readn(E);

```

```
Write ('қан тобы:'); Readln(D);
Writeln(A:2,C:6, B:6, E:8, D:10);
End;
Readln END.
```

3. Карточкадан қан тобы бірдей болатын 10 сынып оқушыларының тізімін шығару программасы.

```
TYPE Dosye= RECORD
A, C: String;
B, E :Integer;
D: Char;
End.

VAR X: Array[1..10] Of Dosye;
K,N: Byte; G: Char;
BEGIN
For K:= 1 To 3 Do With X[K] Do Begin
Write('Фамилиясы: '); Readln(A);
Write('Аты:'); Readln(C);
Write('Туған жылы:'); Readln(B);
Write('Класы:'); Readln(E);
Write ('қан тобы:'); Readln(D);
End; Writeln;
Write('Қандай қан тобы керек: '); Readln(G);
N:=0; Writeln('Осы топқа жататын оқушылар тізімі:');
For K:=1 To 3 Do With X[K] Do Begin
If(D=G) And (E=10) Then Writeln(A:10, C:10,)
Else N:=N+1; If N=3 Then Writeln('Ондай оқушы жоқ')End;
Readln END.
```

4. Көпмүшенің ұқсас мүшелерін біріктіру.

$$4*X+8*Y+5*X+4*C+6*X+5*Y+12*C$$

```
PROGRAM Jazy 2;
TYPE Kob= RECORD
K :Integer;
B: Char;
End.
VAR N,M: Kob; S: Integer;
BEGIN S:=0;
While Not Eoln Do
Begin Readln(N.K); Readln(N.B);
If N.B = 'X' Then S:= S+N.K End;
M.K:=S;
M.B:='X';
Writeln('көпмүшенің X-і бар мүшелері=', M.K, M. B);
Readln END.
```


5. Футбол командаларының жеңіске жеткен және тең ойындар сандары арқылы алған ұпайларын шығару.

Командаларға әр жеңісі үшін 3 ұпай, әр тең ойыны үшін 1 ұпай есептеледі.

A) 1 түрі:

```
PROGRAM Jazy 3;
TYPE Futbool = RECORD
  A: Array [1..17] Of String;
  B, C: Array [1..17] Of Byte;
  D: Array [1..17] Of Integer;
End;
VAR X,Y: Futboll; K: Byte;
BEGIN
For K:=1 To 5 Do Begin
Read(X.A[k]); Readln(X.B[k], X.C[k]);
X.D[k]:=3*X.B[k]+X.C[k]; End;
For K:= 1 To 5 Do Writeln(X.A[k],X.D[k]);
Readln END.
```

ә) 2 түрі:

```
PROGRAM Jazy 4;
TYPE Futboll = RECORD
  A: Array [1..17] Of String;
  B, C: Array [1..17] Of Byte;
  D: Array [1..17] Of Integer;
End;
VAR X: Futboll; K: Byte;
BEGIN
With X Do Begin
For K:=1 To 5 Do Begin
Read(A[k]); Readln(B[k], C[k]);
D[k]:=3*B[k]+C[k]; End;
End;
For K: 1 To 5 Do Writeln(X.A[k],X.D[k]);
Readln END.
```

б) 3 түрі

```
PROGRAM Jazy 5;
TYPE Futboll = RECORD
  A: Array [1..17] Of String;
  B, C: Array [1..17] Of Byte;
  D: Array [1..17] Of Integer;
End;
VAR X: Futboll; K: Byte;
BEGIN
With X Do Begin
For K:=1 To 5 Do Begin
Read(A[k]); Readln(B[k], C[k]);
D[k]:=3*B[k]+C[k]; End;
For K:= 1 To 5 Do Writeln(A[k],D[k]);
```

End;
Readln END.

6. Жоғарыдағы программаның орындалу нәтижесінде экранға командаларды алған орындары бойынша шығару.

```
PROGRAM Jazy ;
TYPE Futboll = RECORD
    A: Array [1..17] Of String;
    B, C: Array [1..17] Of Byte;
    D: Array [1..17] Of Integer;
End;
VAR X,Y: Futboll; K,I: Byte; P: Integer; M: String;
BEGIN With X Do Begin
For K:=1 To 5 Do Begin
Read(A[k]); Readln(B[k], C[k]);
D[k]:=3*B[k]+C[k]; End;
For K: = 1 To 4 Do
For I:= K+ 1 To 5 Do If D [I] > D[k] Then
Begin P := D[K]; D [K] := D[I]; D[I]:=P;
M:=A[K]; A[K]: = A[I]; A[I]: = M End;
For K:= 1 To 5 Do Writeln (K, 'орында : ', A[K],', ' , D[K], 'ұпай ' );
End;
Readln END.
```

Бақылау сұрақтары мен тапсырмалар.

1. Жазулар не үшін қолданылады?
2. Жазулардың сипатталуы.
3. Жазулар өрісі дегеніміз не?
4. Жазулардың программалық тексте қалай жазылады?
5. With операторы қандай қызмет атқарады?
6. Жазу ішіндегі жазулар дегеніміз не?
7. Жазу өрісі компоненттері оқушыларының аты-жөні, туған жылы, адресі, хоббиі, жақсы көретін пәндері, және т.б. деректерден тұратын программа құрыңыз.
8. Жоғарыдағы программаны экранға оқушының аты жөні бойынша барлық деректер шығатындай етіп өзгертіңіз, мысалы:
Әскербек Дархан:
1986 жылы 16 желтоқсанда туылған.
Желтоқсан көшесі 239, тел.3-86-68
Футбол, күрес, шахмат, тоғызқұмалақ.
Математика, еңбек, орыс тілі.
9. Жазу өрісі компоненттері әр жылдары қалалық және облыстық олимпиадаларда жүлделі орындарға ие болған оқушылардың аты-жөнінен, пән аттарынан, жылдары мен алған орындарынан тұратын программа құрыңыз.
10. Жазу өрісі компоненттері мектеп кластарынан, күн және пән аттарынан тұратын "Сабақ кестесі" программасын құрыңыз.
11. Жоғарыдағы программаны күн аттары бойынша экранға сол күні болатын сабақтар шығатындай етіп өзгертіңіз.

12. Жазу өрісі элементтері Менделеев таблицасындағы металдардың қасиеттерінен(аты, реттік нөмірі, атом массасы, ядросының заряды т.б.) тұратын программа құрыңыз.

13. Жоғарыдағы программаны экранға металл аты бойынша барлық деректер шығатындай етіп өзгертіңіз.

14. Жазу өрісі мектеп бітірушілердің аты-жөндерінен, емтихан болатын пәндерден, оқушылардың сол пәндерден алған бағаларынан тұратын программа құрыңыз.

15. Жоғарыдағы программаны оқушының аты бойынша барлық пәндерден алған бағаларын шығаратындай етіп өзгертіңіз.

16. Жазу өрісі 10-ә класс оқушыларының аты-жөні, пән аттары, сол пәндерден алған тоқсандық бағаларынан тұратын программа құрыңыз.

17. Жоғарыдағы программаны орындалу нәтижесінде экранға кластың тоқсандық орташа үлгірімі шығатындай етіп өзгертіңіз.

18. Жазу өрісі 10-а класс оқушыларының аты-жөні, пән аты және сол пәндерден алған жылдық бағаларынан тұратын программа құрыңыз.

19. Жоғарыдағы программаны орындалу нәтижесінде экранға 10 класты үздік бітірген оқушылардың аттары шығатындай етіп өзгертіңіз.

20. Жазу өрісі 10-б класс оқушыларының аты-жөні, туған жылы, айы және күндерінен тұратын программа құрыңыз.

21. Жоғарыдағы программаны орындалу нәтижесінде экранға 1986 жылдың желтоқсан айында туылған оқушылардың аттары шығатындай етіп өзгертіңіз.

22. Жазу өрісі 10-в класс оқушыларының аты-жөні, туған жылы, бойы, салмақтарынан тұратын программа құрыңыз.

23. Жоғарыдағы программаны орындалу нәтижесінде экранға бойы ең ұзын және салмағы ең жеңіл оқушылар аттары шығатындай етіп өзгертіңіз.

№6 Зертханалық жұмыс

Тақырыбы: Көмекші программалар. Көмекші программалар және олардың құрылымы.

Мақсаты:

Турбо Паскальда қолданылатын көмекші программа құрылымы. Процедура ұғымымен танысу.

Теориялық түсініктемелер

6.1. Көмекші программалар және олардың құрылымы

Программалау барысында көптеген бір типті есептеулерге бірнеше бір-біріне ұқсас программа фрагменттерін қайта-қайта жазуға тура келеді. Осындайда артық жұмыс жасамай, барлық есептеулерді бір ғана фрагмент ішінде орындау үшін, көмекші программалар-процедуралар мен функциялар қолданылады. Мысалы программа орындау барысында бірнеше рет ұқсас есептеулер кездессе, онда есептеу блогын бір ақ рет көмекші программаға жазып қойып, программаның кез келген нүктесінен көмекші программаны шақырып, осы блок арқылы есептеуді шығаруға болады. Сонымен көмекші программалар деп программаның орындалу барысында бірнеше рет шақырылып орындалып тұратын операторлар блогынан тұратын көмекші бөлігін айтады.

Көмекші программалар стандартты және пайдаланушылар көмекші программалары болып екіге бөлінеді. Стандартты функциялар мен процедуралар Turbo Pascal тілінің құрамында тұрақты системалық атпен аталып орындалады. Олаға арнаулы USES пайдаланушылар библиотекасында орналасқан- SYSTEM, CRT, GRAPH, GRAPH3, OVERLAY, PRINTER, TURBO3, TURBO VISION модульдерінің барлық функциялары мен процедуралары жатады. SYSTEM модулінің процедуралары мен функциялары Turbo Pascal программасымен бірге автоматты түрде іске қосылады, сондықтан оны пайдаланушылар

бөлімінде көрсетпесе де болады. SYSTEM модулінен өзге модульдер процедуралар мен функцияларын қолдану үшін программаның басында міндетті түрде олардың аттарын USES пайдаланушылар бөлімінде жазу керек, мысалы:

PROGRAM Komek;

USES Graph, Crt;

// Graph, Crt модульдеріндегі стандартты процедуралар мен функцияларды пайдалану үшін ашу.

Стандартты функциялар мен процедуралардың көбісі бұдан бұрынғы бөлімдерде программалар құруда пайдаланылды, мысалы:

Abs(X); //X-тің абсолюттік шамасын есептеу функциясы,

Sin(X); //X-тің синусын есептеу функциясы,

Random(X); //0-ден X-ке дейінгі кездейсоқ сан шығару,

Delay(X); //программаның орындалуын X мс уақытқа тоқтату

Clrscr; // экранды тазалау процедуралары,

Halt; // программаның орындалуын тоқтату процедурасы,т.б.

Пайдаланушылар көмекші программалары деп программистің өзі құрып, аты да өзіне ыңғайлы бір идентификатормен белгіленетін процедуралар мен функцияларды атайды. Олар программаның VAR бөлімінен кейін PROCEDURE немесе FUNCTION қызметші сөздерімен басталып құрылады:

Көмекші программа құрылымы:

Көмекші программа қызметші сөзі көмекші программа тақырыбы;
Көмекші программада кездесетін идентификаторларды сипаттау бөлімдері;
Көмекші программа денесі.

Көмекші программалардың жазылу тәртібі де негізгі программа сияқты бірнеше бөлімдерден тұруы мүмкін, мысалы:

а) Көмекші программа аты, параметрлері жазылатын тақырыбы;

ә) Тұрақтылар бөлімі;

б) Типтер бөлімі;

в) Айнымалылар бөлімі;

г) Операторлардан тұратын программа денесі.

Көмекші программаның аты идентификатормен белгіленеді. Программада бірнеше процедура немесе функция болуы мүмкін. Сол сияқты, көмекші программаның да өзінің көмекші программасы болуы мүмкін. Оларды ішкі программалар деп атайды. Негізгі программада сипатталған идентификаторларды көмекші программаның ішінде де сол сипатталған типпен қолдануға болады. Олармен белгіленген айнымалылар глобальды деп аталады. Ал көмекші программада сипатталған идентификаторды тек өзінің ішкі программасында ғана қолдануға болады. Ол айнымалыларды локальды (жергілікті) деп атайды.

Көмекші программа өздігінен орындалмайды, оның орындалуы үшін негізгі программада көмекші программаның аты мен параметрлерін жазып шақырады, ол жерді көмекші программаны шақыру нүктесі деп атайды. Көмекші программалардың орындалу саны шақыру нүктелерінің санына байланысты болады. Егер негізгі программада бірде бір шақыру нүктесі болмаса, онда көмекші программа орындалмайды. Программа орындалу барысында алдымен негізгі программа орындалады да, керек кезде көмекші программа шақырылады. Көмекші программаны шақыру нәтижесінде негізгі программа уақытша тоқтайды да, басқару көмекші программаға барады. Көмекші программа денесі, негізгі программадан алған деректерді толық өңдеп, орындалғаннан кейін, басқаруды шақыру нүктесіне өңделген жаңа

деректермен бірге қайтарып береді де, негізгі программа орындалуын әрі қарай жалғастырады.

Негізгі программадан деректер қабылдау және көмекші программадан өңделіп жаңарған деректерді негізгі программаға қайтару көмекші программалар параметрлері көмегімен атқарылады. Процедуралар мен функциялардың нақты және формальды параметрлері болады. Программа орындалу барысында есептелетін параметрлер нақты деп, ал көмекші программалардың тақырыбында көрсетілетін параметрлер формальды деп аталады. Нақты және формальды параметрлердің типі, саны, реттілігі сәйкес болуы тиіс. Олар бірдей белгіленуіде мүмкін. Нақты параметрлер тұрақтылардан, айнымалылардан және өрнектерден тұруы мүмкін. Кей жағдайларда көмекші программаның формальды параметрлерін көрсетпесе де болады.

6.2. Процедуралар.

Процедура деп белгілі бір идентификатормен аталып, сол атпен программаның кез келген жерінен шақырып, белгілі бір есептеулерді орындауға болатын программаның тәуелсіз бөлігін айтады. Процедураның құрылымы программа құрылымына ұқсас келеді.

Процедураның құрылымы.

```
PROCEDURE Аты (формальды параметрлер: типтері);  
(*Тақырыбы*)  
Сипаттамалар бөлімдері; (*бұл бөлімдер болмауы да мүмкін*)  
BEGIN  
Операторлар ; (*процедура денесі*)  
END;
```

Программада процедураны пайдалану үшін процедураның атын нақты параметрлерімен бірге жазып шақырады, ол жерді процедураны шақыру нүктесі деп атайды. Программа процедураның шақыруын оқыған заматта процедураның орындалуы басталады. Мысалы программа тақырыбы төмендегідей болса:

PROCEDURE SUMMA (N,M,X:INTEGER);

Онда процедураны программада мынадай жолмен шақыруға болады:

SUMMA (19,A,50);

жақшадағы 19,A,50 нақты параметрлер болып табылады. Нәтижесінде процедурадағы формальды параметрлер мынадай мәндерге ие болады.

N=19, M=A, X=50.

Процедура шақыру процесінде формальды параметрлердің мәндері нақты параметрлердің мәнін қабылдайды да процедура денесіндегі есептеулерде сол жаңа мәндер пайдаланылады.

Процедураның параметрлері мәндік және айнымалы болып, екіге бөлінеді. Мәндік (аргументтік) параметрлер тек қана нақты параметрлердің мәнін өзіне қабылдайды да, қайтып оларға өзінің мәнін бере алмайды. Ал айнымалы параметрлер нақты параметрлердің мәнін қабылдап, олардың процедураның орындалу барысында өзгертіп, нақты параметрлерге жаңа мәндерін қайтып береді. Процедураның бір орындалу барысында негізгі программаның бірден бірнеше идентификаторлары жаңа мән қабылдай алады. Айнымалы параметрлердің алдына VAR қызметші сөзі қойылып жазылады. Мысалы:

```
PROCEDURE Misal (A,B:INTEGER; VAR X,Y:LONGINT);
```

VAR S: INTEGER;

BEGIN S:=A+5*B; X:=SQR(S); Y:=SQR(X); END;

Бұл процедураны программалық тексте төмендігідей жазып шақыруға болады.

Misal(10, 5, K, I)

Программа орындалу барысында Misal(10,5,K,I) шақыру идентификаторын оқып, басқаруды Misal процедурасына жібереді де, төмендегідей әрекеттер орындалады:

1.Процедураның A, B, X, Y формальды параметрлері 10,5,K, I нақты параметрлердің мәнін қабылдайды.

A=10, B=5, X=K, Y=I

2. Процедура денесіндегі операторлар орындалады:

S=10+5*5=35, X=35*35=1225,

Y=1225*1225=1500625;

3. VAR қызметші сөзінен кейін жазылған формальды X және Y параметрлері процедураның орындалу барысында меншіктеген өздерінің жаңа мәндерін нақты K мен I параметрлеріне бергеннен кейін, басқару Misal(10,5,K,I) шақыру идентификаторынан кейін тұрған операторға өтеді;

K=X=1225, I=Y=1500625.

4. Әрі қарай программаның есептеулерінде K және I айнымалылары жаңа меншіктеген мәндерімен қатысады.

6. 3. Процедура құру мысалдары.

1. Катеттері бойынша үшбұрыштың ауданы мен гипотенузасын табу процедурасын құру

Дұрыс құрылған процедуралар мысалдары:

а)

PROGRAM Proced1;

VAR A, B, C, S,: Real;

PROCEDURE Ushbur(M,N: Real; VAR T,Q: Real);

BEGIN

T:=M*N/2; Q:=Sqrt(Sqr(M)+Sqr(N)); END;

BEGIN Readln (A,B);

Ushbur (A,B,S,C); {негізгі программадан Ushbur процедурасын шақыру}

Writeln ('gip=',C:4:2); Writeln ('aud=',S:4:2);

Readln END.

ә)

PROGRAM Proced2;

VAR A, B, C, S,: Real;

PROCEDURE Ushbur(M,N: Real; VAR T: Real; VAR Q: Real);

BEGIN

T:=M*N/2; Q:=Sqrt(Sqr(M)+Sqr(N)); END;

BEGIN Readln (A,B);

Ushbur (A,B,S,C);

Writeln ('gip=',C:4:2); Writeln ('aud=',S:4:2);

Readln END.

б)

PROGRAM Proced3;

VAR A, B, C, S,: Real;

PROCEDURE Ushbur;

```

BEGIN
S:=A*B/2; C:=Sqrt(Sqr(A)+Sqr(B)); END;
BEGIN Readln (A,B);
Ushbur;
Writeln ('gip=',C:4:2); Writeln ('aud=',S:4:2);
Readln END.

```

Дұрыс құрылмаған процедуралар мысалдары:

а)

```

PROGRAM Proced4;
VAR A, B, C, S,: Real;
PROCEDURE Ushbur(M,N: Real; T,Q: Real);
BEGIN
T:=M*N/2; Q:=Sqrt(Sqr(M)+Sqr(N)); END;
        {бұл процедура S-пен C-ға жаңа мән қайтармайды}
BEGIN Readln (A,B);
Ushbur (A,B,S,C);
Writeln ('gip=',C:4:2); Writeln ('aud=',S:4:2);
Readln END.

```

ә)

```

PROGRAM Proced5;
VAR A, B, C, S,: Real;
PROCEDURE Ushbur(A,B: Real; VAR S: Real; C: Real);
BEGIN
S:=A*B/2; C:=Sqrt(Sqr(A)+Sqr(B)); END;
        {бұл процедура S-ке жаңа мән қайтарады, ал C ға қайтармайды}
BEGIN Readln (A,B);
Ushbur (A,B,S,C);
Writeln ('gip=', C:4:2); Writeln ('aud=',S:4:2);
Readln END.

```

б)

```

PROGRAM Proced6;
VAR A, B,C,S,: Real;
PROCEDURE Ushbur(S,C: Real; A,B: Real);
BEGIN
S:=A*B/2; C:=Sqrt(Sqr(A)+Sqr(B)); END;
        {бұл процедура параметрлері реттеліп жазылмаған}
BEGIN Readln (A,B);
Ushbur (A,B,S,C);
Writeln ('gip=',C:4:2); Writeln ('aud=',S:4:2);
Readln END.

```

2. Екі санның үлкенін табатын процедура.

```

PROGRAM Maxsan;
VAR A, B, Max: Real;
PROCEDURE Maxim(K,I: Real; VAR S: Real); {Maxim процедурасы }
BEGIN If K>I Then S:=K
Else S:=I; End;      {екі санды салыстырып үлкенін табу}
BEGIN Write ('2 сан енгіз:');Readln(A,B); {негізгі программа}
Maxim(A,B,Max); Writeln('Max=',Max:4:1); Readln End.

```

ә) Екі санның үлкенін табатын процедураны қолданып, берілген үш және төрт санның үлкенін таду:

```
PROGRAM Maxsan2;
VAR A, B, C, D, Max: Real;
PROCEDURE Maxim(K, I: Real; VAR S: Real); {Maxim процедурасы }
BEGIN If K>I Then S:=K Else S:=I; End;
BEGIN Write ('4 сан енгіз'); Readln(A, B, C, D); {негізгі программа}
Maxim(A, B, Max); {процедураның орындалуы нәтижесінде Max A мен B
сандарының үлкенін меншіктейді}
Maxim(Max, C, Max); {процедураның орындалуы нәтижесінде Max A, B, C
сандарының үлкенін меншіктейді}
Writeln('A, B, C сандарының үлкені: ', Max: 4: 1)
Maxim(Max, D, Max); {процедураның орындалуы нәтижесінде Max A, B, C, D
сандарының үлкенін меншіктейді}
Writeln('A, B, C, D сандарының үлкені: ', Max: 4: 1); Readln End.
```

3. N-нен M-ге дейінгі бүтін сандар қосындысын табу процедурасы.

```
PROGRAM Proc7;
VAR N, M, K: Integer; S: Longint;
PROCEDURE Sum (N, M: Integer; VAR X: Longint);
{Сандардың енгізу процедурасы}
Begin X:=0; For K:=N To M Do X:=X+K; End;
BEGIN Readln(N, M); Sum(N, M, S); Writeln ('SUM=', S);
Readln END.
```

4. Бүтін X санының бүтін Y дәрежесін табу процедурасын құру.

```
PROGRAM Proc8;
VAR X, Y: Integer; S: Longint;
PROCEDURE Inp (VAR A, B: Integer); {Сандардың енгізу процедурасы}
Begin Write ('Санды енгізу: '); Readln(A); Write('Дәрежесін енгізу: ');
Readln (B) End;
PROCEDURE Dar (A, B: Integer; VAR S: Longint);
VAR K: Byte; {Дәреже есептеу процедурасы}
Begin S:=1; For K:=1 To B Do S:=S*A; End;
PROCEDURE Out(S: Longint); {Нәтиже шығару процедурасы}
Begin Writeln (X, 'санының', Y, 'дәрежесі=', S)End;
BEGIN Inp(X, Y); Dar(X, Y, S); Out(S); Readln END.
```

5. N санының цифрларының санын шығару процедурасы.

```
PROGRAM Proc8;
VAR N: Longint;
PROCEDURE San; {San процедурасы}
VAR K: Integer;
Begin K:=0; {санның цифрларын есептеу циклі:}
Repeat
K:=K+1; N:=N Div 10; Until N=0;
Write ('N санының цифрларының саны=', K) End;
```


BEGIN Readln (N); San; Readln End. {негізгі программа}

6. N санының цифрларының жұп-тақтығын анықтайтын процедурасын құру.

```
PROGRAM Proc9;
VAR N: Longint; K,I:Integer;
PROCEDURE San(I: Longint);      { San процедурасы }
Begin K:=0;      { санның цифрларын есептеу }
Repeat K:=K+1;
I:=I Div 10; Until I=0; End;
PROCEDURE Dar (I:Integer;VAR P: Longint); {Dar процедурасы}
VAR J: Integer; {оның I дәрежесін есептеу}
Begin P:=1; For J:=1 To I do P:=P*10; End;
PROCEDURE Jup (N : Longint);    {Jup процедурасы}
VAR X,D: Longint;
Begin {санының цифрларының жұп-тақтығын анықтайтын цикл}
For I:=1To K do Begin
Dar (K-I,D); {D идентификаторы Dar процедурасының орындалу нәтижесінде
              10-ның K-I дәрежесін меншіктейді}
X:=N Div D; {X идентификаторы I=1 болғанда санның 1-ші цифрын, I=2 болғанда-
              2-ші цифрын, ..., I=K болғанда K цифрын меншіктейді}
Write (I,'цифр=',x);
N:=N Mod D;      {келесі циклге қалатын санның шамасы}
If X Mod 2=0     {цифрлардың жұп тақтығын салыстыру}
Then Write ('бұл цифр жұп')
Else Write ('бұл цифр так'); Writeln; End; End;
BEGIN Readln (N); San (N); Jup(N); {негізгі программа}
Readln End.
```

7. Массив элементтерін өндіру, шығару процедурасын құру мысалдары.

а) массив элементтерінің қосындысы мен көбейтіндісін табу:

```
PROGRAM Massiv1;
CONST N=5;
TYPE Mas=Array[1..N] Of Integer;
VAR A,B:Mas; S,P,K : Integer;
PROCEDURE Inp (N: INTEGER; VAR X:MAS);
BEGIN For K:=1 To N Do Read (X[K]);END;
PROCEDURE Out (N: Integer; VAR X:Mas; VAR S,P :Integer);
BEGIN S:=0; P:=1; For K:=1 To N Do Begin S:=S+X[K];
P:=P*X[K] End End;
BEGIN Inp (N,A); Out(N,A,S,P); {негізгі программа}
Writeln ('Қосынды=',s); Writeln ('Көбейтінді=',P);
Inp (N,B); Out(N,B,S,P); Writeln ('Қосынды=',S);
Writeln ('Көбейтінді=',P); Readln End.
```

ә) массив элементтерінің қосындысы мен көбейтіндісін табу процедурасын параметрлерсіз құру:

```
PROGRAM Massiv2;  
CONST N=5;  
VAR X,A,B: Array[1..N]Of Integer; S,P,K : Integer;  
PROCEDURE Inp;  
Begin For K:=1 To N Do Read (X[K]); Writeln End;  
PROCEDURE Out;  
Begin S:=0; P:=1; For K:=1 To N Do  
Begin S:=S+X[K]; P:=P*X[K] End End;  
BEGIN Inp; Out;      { негізгі программа }  
Writeln ('Қосынды=',S); Writeln ('Көбейтінді=',P);  
Inp; Out;  
Writeln ('Қосынды=',S); Writeln ('Көбейтінді=',P);  
Readln END.
```

№7 Зертханалық жұмыс

Тақырыбы: Функциялар.

Мақсаты:

Турбо Паскальда қолданылатын көмекші программа құрылымы. Функция және рекурсиялар ұғымдарымен танысу.

Теориялық түсініктемелер

Функциялардың атқаратын қызметі мен жұмыс принциптері процедураларға ұқсайды. Функцияның процедуралардан негізінен екі айырмашылықтары бар:

1. Функциялар өрнек операндалары ретінде қолданыла береді;
2. Функция шақыру нүктесіне бір ғана өзінің атына меншіктелген скалярлық шаманы қайтарады. Процедуралар шақыру нүктесіндегі идентификаторға жаңа мәндер қайтарып бере алса, ал функциялар бір шақыру нәтижесінде бір ғана идентификаторға жаңа мән бере алады.

Функцияның құрылымы:

```
FUNCTION аты (формальды параметрлер: типтері): типі;  
Сипаттамалар бөлімдері; (*бұл бөлімдер болмауы да мүмкін)  
BEGIN Операторлар бөлімі;  
Аты:=(*функция жаңа мән меншіктейді*) (*функция денесі*)  
END;
```

Функциялар да процедуралар сияқты шақырылады. Негізгі программаға жаңа мән беру үшін функция денесінде функцияның идентификаторы кемінде бір мән меншіктеуі тиіс. Шақыру нүктесіне функцияның соңғы меншіктеген нәтижесі жаңа мән болып қайтарылады.

Функция құру мысалдары.

1. Екі $A(x_1, y_1)$ және $B(x_2, y_2)$ нүктелерінің координаталары бойынша ара қашықтығын анықтау функциясын құру.

```
PROGRAM Func1;  
VAR X1,X2,Y1,Y2: Integer;  
FUNCTION Ara:Real;      { қашықтық табу функциясы }  
Begin Ara:=Sqrt(Sqr(X2-X1)+Sqr(Y2-Y1)) End;  
BEGIN Readln (X1,X2,Y1,Y2);      { негізгі программа }  
Writeln ('екі нүктенің ара қашықтығы =', Ara:4:1);  
Readln END
```

2. Нақты оң X санының нақты Y дәрежесін табу функциясын құру.

```
PROGRAM Func 1;  
USES Crt;  
VAR X,Y: Real;  
FUNCTION Dar (A,B:Real):Real; {дәреже табу функциясы}  
Begin If A>0 Then Dar :=Exp(Ln(A)*B)  
Else Begin Writeln (`ter`); Delay (30000); Halt End End;  
BEGIN Clrscr; Readln (X,Y); {негізгі программа}  
Writeln (`dar =`, Dar(X,Y):4:1); Readln END
```

3. M-нен N-ге дейінгі бүтін сандар қосындысын табу функциясы.

```
PROGRAM Func 3;  
VAR A,B,P:Integer;  
FUNCTION Sum (M,N: Integer): Integer;  
Var s,k:integer;  
Begin S:=0; For K:=M To N DO S:=S+K;  
Sum:=S; End;  
BEGIN Readln (A,B);  
P:=Sum(A,B);  
Writeln (`P=`P); Readln END.
```

4. N санының факториалын табу функциясы.

```
PROGRAM Func 2;  
VAR N,P,K:Integer; S:Longint;  
FUNCTION Fact(M:Integer):Longint;  
Begin P:=1; For K:=2 To M Do P:=P*K;  
Fact:=P; End;  
BEGIN Repeat Readln (N);Until N>0 {оң сан ендіру циклі}  
S:=FACT(N);  
Writeln (`Fac=`S); Readln END.
```

5. Екі санның және төрт санның кішісін табу функциялары.

```
PROGRAM Minsan;  
VAR A,B,C,D:Real;  
FUNCTION Min(K,I:Real):Real;  
Begin If K<I Then Min:=K Else Min:=I; End;  
FUNCTION Min4(A,B,C,D:Real):Real;  
Begin Min4:=Min(D, Min(C, Min(A,B))); End;  
BEGIN Writeln(`Төрт сан енгіз=`); Readln (A,B,C,D);  
Writeln(`Төрт санның кішісі=`); MIN4 (A,B,C,D):4:2);  
Readln END.
```

6. Кез-келген жолдық қатардың латын алфавитінің кіші әріптерін бас әріпке айналдыру функциясын құру.

```
PROGRAM Kat3;  
VAR X:String; K:Byte;  
FUNCTION Basari p(A: String): String;  
Begin For K:=1 To Length(A) Do A[K]:=Uppcase(A[K]);  
Basari p:=A End;  
BEGIN Readln (X); X:=Basari p(X);  
Writeln (`Қатардың жаңа мәні X=`X); Readln END.
```

7. X жолдық қатарында кездесетін `а` символдарын санап, оларды `b` символымен ауыстыру функциясын құру.

```
PROGRAM Kat4;
VAR X:String; K,N:Byte;
FUNCTION Austyr (A: String): String;
Begin N:=0 For K:=1 To Length (A) Do If A[K]=`a` Then
Begin N:=N+1; A[K]:=`b` End; Austyr:=A End;
BEGIN Readln (X); X:= Austyr (X)
Writeln (Қатардың жаңа мәні X=`X` қатарда `N`,`a` символы болды`);
Readln END.
```

8. 2-ден 1000-ға дейін сол жағынан да, оң жағынан да оқығанда жай сандарды анықтау функциясын құру.

```
PROGRAM Func5;
VAR X,N: Integer;
FUNCTION Jai(Y: Integer):String;
VAR D,I: Integer; T:String;
Begin T:= `Жай сан емес `; D:=0;
FOR I:=2 TO Y-1 DO If Y Mod I=0 Then D:=D+1;
If D=0 Then T:=`Жай` ; Jai:=T END;
FUNCTION Keru(Y:INTEGER ):integer;
VAR S:integer;
Begin if Y<10 Then S:=Y;
If(Y>10) And (Y<100) Then S:=Y Mod 10*10+Y Div 10;
If(y>=100) And (Y<1000)
Then S:=Y Mod 10*100+Y Mod 100 Div 10*10+Y Div 100;
Kery:=S; End;
BEGIN Writeln( Бұндай сандар`); N:=2;
For X:=4 To 99 Do Begin
If Jai(X)=`Жай` Then If Jai(Keru(X))=`Жай`
Then Begin Writeln(X); N:=N+1 End; End;
Writeln (N, `жай сан бар`);
Readln End.
```

9. Тік төртбұрышты квадраттарға бөлу программасының функциясы.

```
PROGRAM Kvatrat;
VAR A,B,D,K,X,Y: Integer;
FUNCTION Min(A,B: Integer): Integer;
Begin If A<B Then Min:=A Else Min:=B; End;
FUNCTION Max(A,B: Integer): Integer;
Begin If A>B Then Max:=A Else Max:=B; End;
BEGIN Readln (A,B);
X:=Min(A,B); Y:=Max(A,B); K:=1;
While X<>Y Do Begin K:=K+1;
D:=Y-X; Y:=Max(D,X); X:= Min(D,X); End;
Writeln (Қвадраттар саны = `K);
Readln END.
```

Рекурсиялар.

Көмекші программалардың денесіндегі операторлардың орындалу барысында өзін - өзі шақыру процессі **рекурсия** деп аталады. Рекурсивті программаларда көмекші программаның орындалуы есептеу параметрінің белгілі бір мәнінен бастап тізбектей төменгі деңгейлі мәніне қарай қашан есептің тривиальды нәтижесі алынғанға дейін жүреді. Функцияның әр рекурсивті орындалуында оның алдыңғы айнымалылары программалық ағым деп аталатын жадта сақталады. Рекурсивті функциялар құрғанда программалық ағымның шексіз емес екендігін есте ұстаған жөн. Рекурсияның мысалы ретінде төменгі N санының факториалын табатын функцияны келтіруге болады:

N санының факториалын табу үшін рекурсия құру мысалы:

```
PROGRAM Recurs;  
VAR N: Integer; F: Longint;  
FUNCTION Fact(N: Integer): Longint;  
Begin If N<0 Then Begin;  
Writein(`N -ді дұрыс енгізіңіз `); Fact:=0 End  
Else If N=1 Then Fact:=1  
Else Fact:=N*Fact(N-1); End;  
BEGIN Readln(N); F:=Fact(N);  
Writeln(`FN=` ,F);  
Readln END.
```

Бұл программаның орындалу барысында негізгі программдан Fact функциясы шақырылады. Функцияның денесі N санын 0-мен салыстырудан басталады. Егер сан теріс немесе 0-ге тең болса, онда функция 0-ді меншіктейді де орындалуын тоқтатады. Егер сан 1-ге тең болса функцияның мәні 1-ді меншіктейді де орындалуын тоқтатады. Егер санның мәні 1-ден артық болса, онда басқару $Fact := N * Fact(N-1)$ операторына барады да N-нің мәні 1-ге дейін жеткенше функцияның орындалуы қайталанады (факториалдың тривиальды мәні 1-ге тең болғандықтан $N=1$). Функция рекурсивті процесс тоқтағаннан кейінгі көбейтінді нәтижесін меншіктейді де орындалуын тоқтатады. Мысалы $N=5$ болғанда Fact функциясы төмендегідей орындалады:

```
N=5; 5>1; Fact:=5*Fact(4);  
N=4; 4>1; Fact (4):=4*Fact(3);  
N=3; 3>1; Fact (3):=3*Fact(2);  
N=2; 2>1; Fact (2):=2*Fact(1);  
N=1; 1=1; Fact(1):=1;
```

$Fact(1)=1$ меншіктегеннен кейін рекурсивті процесс тоқтап, программалық ағымдағы орналасқан айнымалылар мәндері біртіндеп есептеліп, функцияның қорытынды нәтижесі шығарылады:

```
Fact(2)=2*Fact(1)=2*1=2;      Fact(3)=3*Fact(2)=3*2=6;  
Fact(4)=4*Fact(3)=4*6=24;    Fact=5*Fact(4)=5*24=120.
```

Рекурсия құру мысалдары.

1. X қатарын кері шығару рекурсиясын құру.

```
PROGRAM Katar;  
VAR X:String;  
FUNCTION Kery(Y: String): String;  
VAR P:Char; { P- қатардың бірінші символы }  
Z: String; { Z – қатардың бірінші символы өшірілгеннен кейінгі  
қалдық }
```

```

Begin If Length(Y)=1 Then Kery:=Y
Else Begin P:=Y[1]; Delete(Y,1,1);
      Z:=Kery(Y);      { функцияны қайта шақыру (рекурсия) }
Kery:=Concat(Z,P) End End;
BEGIN Readln (X);
Writein(`Қатардың кері оқылуы X=` ,Kery(X));
Readln END.

```

2. Тік төртбұрышты квадраттарға бөлу программасының рекурсиясы.

```

PROGRAM Kvatrat;
VAR A,B,D,K,X,Y: Integer;
FUNCTION Min(A,B: Integer): Integer;
Begin If A<B Then Min:=A Else Min:=B; End;
FUNCTION Max(A,B: Integer): Integer;
Begin If A>B Then Max:=A Else Max:=B; End;
FUNCTION Kvad: Integer;
Begin If X=Y Then Kvad:=1 Else Begin D:=Y-X; Y:=Max(D,X);
X:=Min(D,X); Kvad:=Kvad+1; End End;
BEGIN Readln (A,B);
X:=Min(A,B); Y:=Max(A,B);
Writeln (`Квадраттар саны =`, Kvad);
Readln END.

```

Бақылау сұрақтары мен тапсырмалар.

1. Көмекші программалар дегеніміз не, олардың атқаратын қызметі?
2. Көмекші программалардың түрлері, жазылу тәртібі мен құрылымы.
3. Қандай стандартты процедуралар мен функцияларды білесіз?
4. Процедуралар дегеніміз не және оның құрылымы.
5. Шақыру нүктесі дегеніміз не?
6. Функциялар дегеніміз не және оның құрылымы?
7. Процедуралар мен функциялардың айырмашылығы неде?
8. Нақты және формальды параметрлер дегеніміз не?
9. Рекурсия дегеніміз не?
10. Бірнеше сандардың арифметикалық ортасын табу процедурасын құрыңыз.
11. А мен В интервалында жататын барлық жай сандарды табатын процедура құрыңыз.
12. Екі $A(X_1, Y_1)$ және $B(X_2, Y_2)$ нүктелерінің ара қашықтығын процедура құрып табыңыз.
13. Кез-келген көпбұрыштың төбелерінің координаттары арқылы олардың периметрін табу процедурасын құрыңыз.
14. Кез-келген интервалында кездейсоқ сан шығаратын процедура құрыңыз.
15. Жер экваторының ұзындығын есептейтін процедура құрыңыз.
16. Екілік санақ жүйесінде жазылған санды ондық санақ жүйесіндегі санға айналдыру функциясын құрыңыз.
17. А және В интервалында жататын барлық палиндром сандар қосындысын табатын функция құрыңыз.
18. А және В интервалында жататын барлық жай сандар мен кемел сандардың қосындысын табатын функция құрыңыз.
19. Массивтің жұп элементтерінің санын табатын функция құрыңыз.
20. Массивтің ең үлкен тақ элементін табатын функция құрыңыз.
21. Матрицаның бас диагональ бойында орналасқан элементтерінің қосындысын табу функциясын құрыңыз.

22. Матрицаның элементтерінің орналасуын сағат тілімен 90 градусқа бұру функциясын құрыңыз.

23. Төмендегі дәреже табуға арналған рекурсивті функцияның орындалуына талдау жасаңыз.

```
PROGRAM Rekurs;  
VAR K,X,Y:Integer;  
Function Dar(N,M:Integer):Longint;  
VAR P:Longint;  
Begin If M=1 Then P:=N  
Else P:=N*Dar(N,M-1);  
Dar:=P End;  
Begin Readln(X,Y);  
Writeln('X^Y=',Dar(X,Y)); Readln END.
```

№8 Зертханалық жұмыс

Тақырыбы: Файлдар.

Мақсаты:

Турбо Паскальда қолданылатын файлдар. Файлдардың сипатталуы және олардың процедура мен функция арқылы сипатталуын түсіндіру.

Теориялық түсініктемелер

Файлдар деп сыртқы жад құрылғыларына (дискілерге) белгілі бір атпен жазып қоятын кез-келген мәліметтер жиындығын айтады. Программаның өзі де файл болып табылады. Кейбір өте үлкен программаларда мәліметтер қайта-қайта өңдеу программалаушының ұзақ уақытын алады, және оны шаршатып жібереді де жұмыс барысында көптеген қателердің жіберілуіне әкеп соғады. Сондықтан көпмәліметтерді енгізу мен шығаруды қажет ететін программаларды файлдарға жазып, дискілерге сақтап қойған жұмысты көп жеңілдетеді. Turbo Pascal –да файлдар бір типті элементтердің тізбегінен тұрады, олардың негізгі 3 түрлі ерекшеліктері болады:

- оның белгілі бір аты болады, бұл бір мезгілде бірнеше файлдармен жұмыс істеуге мүмкіндік береді;
- оның компоненттерді бір типті болады, мысалы жазулардан немесе жолдық қатарлардан тұратын файл жасауға болады;
- бір файлдың ішінде файл жасауға болмайды, файлдардан тұратын файл жасауға болмайды; Файлдарға қолданылатын амалдар.

Файлдан берілген қатарды іздеп шығару:

```
Program Ft3;  
USES Crt;  
VAR X: Text; T:Integer;  
Y,S:String; N:String  
BEGIN Clrscr; Write('Файлдың аты: '); Readln(S); T:=0;  
While Not Eof(X) Do Begin  
Readln(X,Y);  
IF Y=S Then Begin T:=1; Writeln('Y=',Y); End; End;  
If T=0 Then Writeln('Ондай қатар жоқ');  
Readln; Close(X); END.
```

Файл соңына мәлімет ендіру.

```
PROGRAM Ft4;
VAR X:Text; Y,N:String

BEGIN Write('Файлдың аты: '); Readln(N)
Assign(X,N); Append(X);
Repeat Readln(Y); Writeln(X,Y); Until Y='';
Writeln('Файлдың соңы'); Close (X); Readln END.
```

Жаңа файлға текстің бір қатарын көшіру:

```
PROGRAM Ft5
VAR X,X1:Text;Y,N,New:String;

    BEGIN Write('Файл аты: '); Readln(N)
    Assign(X,N); Reset(X);
    Write('Жаңа файл аты: '); Readln(New);
    Assign(X1,New); Rewrite(X1);
    While Not SeekEoln(X) Do Begin
    Read(X,Y); Writeln(x1,Y); Readln END.
```

Дискіде бар файлға мәлімет көшіру:

```
PROGRAM Ft6
Var X,X1: Text; Y,N,New:String;

    BEGIN
    Write('Файл аты: '); Readln(N);
    Assign(X,N); Reset(X)
    Write('қай файлға көшіру керек: '); Readln(New);
    Assign(X1,New); Append(X1);
    While Not Eof(X) Do Begin Readln(X,Y); Writeln(X1,Y); End;
    Close(X); Close(X1); Readln End.
```

Элементтерді бүтін сан болатын текстік файлмен жұмыс.

А) файл жазу

```
USES Crt;
Var X: Text; Y,K:Integer; N:String;
BEGIN Clrscr; Write('Файлдың атын енгіз : '); Readln(N);
Assign(X,Y); Rewrite(X);
For K:=1 to 10 Do Begin
Read(Y0;
Writeln(x,Y); End;
Writeln('файлдың соңы'); Readln; Close(X) END.
```

Ә) файлды оқу:

Program Tf;

```
USES Crt;
VAR X:Text; N:String;
```



```

Y:Integer;
BEGIN Clrscr; Write('Файдың атын енгіз : '); Readln(N);
Assign(x,N); Reset (X);
While Not Eof(X) Do Begin
Readln (X,Y);
Write(' Y=',Y); End;

```

Б) файлдан теріс сандар шығару:

```

Program Ft5;
Uses Crt;
VAR X: Text; Y:Integer; N:String;
Begin Clrscr; write('Файлдың атын енгіз: ');
Readln(N); Assign(X,N); Reset(X);
While Not Eof(x) Do Begin
Readln(X,Y); if Y<0 Then Writeln('Y=',Y); End;
Close (X); Readln END.

```

В) файлдан жұп сандарды шығыру.

```

PROGRAM Tf;
VAR X: Text; Y:Integer ; N:String;

    BEGIN Write('Файлдың атын енгіз: ');
    Readln(N); Assign(X,N); Reset(X);
    While Not Eof(X) Do Begin

        Readln(X,Y); if y Mod 2=0 Then Writeln('Y=',Y); End;          Close (X);  Readln END.

```

Г) файлдың 2-ге де 4-ке де бөлінетін элементтерін шығару:

```

PROGRAM Ft5;
VAR X:Text; Y:Integer; N:String;
BEGIN Write('Файлдың атын енгіз: '); Readln (N);
Assign(X,N);
Rewrite(X);
For K:=100 TO 99 Do Writeln(X,K);
Write('Файлдың соңы: '); Readln; Close(X); End.

```

е) үш таңбалы сандардан тұрғын файлдың жай сандарын басқа бір файлға жазу :

```

PROGRAM Js;
VAR X,X1:Text; K,I,T:Integer; N,N1:String;
BEGIN Write('Файлдың атын енгіз : '); Readln(n);
Assign(X,N);
Reset(x);
Write('gana faildin ati: '); Readln(N1);
Assign(X1,N1); Rewrite(X1);
For K :=100 To 999 Do BeginT:=0;
For I:=2 To k-1 Do If K MOD I=0 Then T:=T+1;
If T=0 Then writeln (X1,K); End;

```

```
Write('Файлдың соңы'); Readln;  
Close(X); Close(X1); End.
```

И) жай сандар жазылған файлды оқу

```
PROGRAM Js;  
VAR X: Text; K:Integer; N:String;  
Begin Write('Файлдың атын енгіз '); Readln(N);  
Assign(X,N);  
Reset(X);  
While Not Eof(X) Do Begin  
Readln(X,K); Write(K=' ',K); End;  
Writeln; Write('Файлдың соңы'); Readln; Close(X); END.
```

Элементтері массив болатын текстік файлмен жұмыс

А) файл жазу:

```
PROGRAM Tf;  
USES Crt;  
VAR X: Text; K:Integer; N:String;  
A:Array[1...5] Of Integer;  
BEGIN Clrscr; Write('Файлдың атын енгіз: '); Readln(N);  
Assign(X,N); Rewrite(X);  
For K:=1 To 5 Do Begin  
Read(A[K]);End;  
Writeln; Writeln('файлдың соңы'); Close(X); End.
```

Ә)Файлды оқу:

```
PROGRAM Tf;  
USES Crt;  
VAR X:Text; K:Integer; N:String;  
  
A:Array[1...5] Of Integer;  
BEGIN Clrscr; Write ('Файлды4 атын енгіз: '); Readln(N);  
Assign(X,N); Reset(X); K:=0;  
While Not Eof(X) Do Begin Inc(K);  
Readln(X,A[K]); Write ('A[' ,K,']=', A[K]); End;  
Writeln; Writeln('Файлдың соңы'); Close(X); Readln End.
```

Б) файлға жазылған массивтің оң элементтерінің қосындысын табу:

```
PROGRAM Tf;  
USES Crt;  
VAR X:Text; N:String;  
K,S:Integer; A:Array[1...5] of Integer;  
BEGIN Clrscr; Write('Файлдың атын енгіз: '); Readln(N);  
Assign(X,N); Reset(X);  
S:=0;K:=0; while Not Eof(X) Do Begin Inc(K);  
Readln(X,A[K]); If A[K]>0 Then S:=S+A[K]; End;
```

```

Writeln('S=',S);
Writeln;Writeln('Файлдың соңы');Close(X)
Readln End.

```

В) Файлға жазылған массивтің жұп элементтерін шығару:

```

PROGRAM Tf;
USES Crt;
Var X:Text; N:String;
K:Integer; A:Array[1..5] of Integer;
Begin Clrscr; Write('Файлдың атын енгіз: ') Readln(N);
Assign(X,N); Reset(X);
K:=0; While Not Eof(X) Do Begin Inc(K);
Readln(X,A[K]);
If A[K] Mod 2=0 Then Write(' A[' ,K, ']=',A[K]); End;
Writeln; Writeln('Файлдың соңы'); Close(X);
Readln End.

```

Г)Файлға жазылған массив элементтерін өсу бойынша реттеу:

```

PROGRAM Tf;
USES Crt;
Var X: Text; N:String;
K,I,P:Integer; A:Array[1..5] of Integer;
Begin Clrscr ; Write('Файлдың атын енгіз: '); Readln(N);
For K:=1 To 4 DO
For I:=K+1 To 5 Do Write('A[' ,K, ']=',A[K]);
Writeln; Writeln('Файлдың соңы'); Close(X); Close(X1); Readln End.

```

Элементтері матрица болатын текстік файлмен жұмыс.

А) файл жазу:

```

PROGRAM Tf;
VAR X: Text; N:String;
A:Array[1..5,1..3] Of Integer; L,I:integer;
BEGIN Write('Файлдың атын енгіз: '); Readln(N);
Assign(X,N); Rewrite(X);
For K:=1 To 5 Do
For I:=1 To 3 Do Begin Read(a[K,I]);
Writeln (X,A[K,I]); End;
Writeln ; Writeln('Файлдың соңы'); Readln; Close(x); END.

```

Ә)файл оқу:

```

PROGRAM Tf;
VAR X: Text; N:String;
A:Array[1..5,1..3] Of Integer; K,I:Integer;
BEGIN Write('Файлдың атын енгіз: '); Readln(N);
Assign(X,N); Reset(X);
For K:=1 To 5 Do Begin
For I:=1 To 3 Do Begin

```

```
Read (X,A[K,I]); Write(A[K,I]:4); End;  
Writeln; End;  
Writeln; Writeln('Файлдың соңы'); Readln; Close(X); End.
```

Б) файлдағы матрицаның теріс элементтерінен бір өлшемді массив шығару:

```
PROGRAM Tf;  
VAR X: Text; N:String;  
A:Array[1..5,1..3] Of Integer;  
B:Array[1..15] Of Integer;  
K,I,T:integer;  
BEGIN Write (' Файлдағы атын енгіз: '); Readln(N);  
Assign(X,N); Reset(X);  
T:=0;  
For K:=1 To 5 Do  
For I:=1 To 3 Do Begin  
Read(X,A[K,I]);IF A[K,I]<0 Then Begin Inc(T); B[T]:=A[K,I]; End; End;  
For K:=1 To T Do Write(' B[',K,']= ',B[K]);  
Writeln; Writeln(' Файлдың соңы ');  
Readln; Close(X); END.
```

В) матрицаның оң элементтерін басқа файлға бір өлшемді массив етіп жазу:

```
PROGRAM Tf;  
USES Crt;  
K,T,I,P:Integer;  
A:Array[1..5,1..3] Of Integer;  
B:Array[1..15] Of Integer;  
Begin Clrscr; Write('Файлдың атын енгіз: '); Readln(N);  
Assign(X,N); Reset(X);  
Write('Жаңа файл: '); Readln(New);  
Assign(X1,New); Rewrite(X1);  
T:=0;  
While Not Eof(X) Do  
For K:=1 To 5 Do  
For I:=1 To 3 Do  
Begin Readln (X,A[K,I]);  
If A[k,i]>0 Then Begin Inc(T); B[T]:=A[K,I]; Writeln(X1,B[T]); End; End; WRITE(X1,T);  
Writeln;Writeln('Файлдың соңы'); Close(X);Close(X1); Readln END.
```

Г) оң элементтер жазылған файлды оқу:

```
PROGRAM Tf;  
VAR X:Text; N:String;  
B:Array[1..15] Of Integer;  
K,T:Integer;  
BEGIN Write('Файлдың атын енгіз: '); Readln(N);  
Assign(X,N); Reset(X);  
While Not Eof(X) Do Readln(X,T);  
Close(X); Reset(X);  
For K:=1 To T Do Begin Readln(X,B[K]); Write('B[',K,']= ', B[K]);End;  
Writeln; Writeln('Файлдың соңы'); Readln; Close(X); End.
```

Д) файлдағы матрицаны басқа файлға қатарларын өсу бойынша реттеп жазу:

```
PROGRAM Tf;  
USES Crt;
```

```

VAR X,X1:Text;N;New:String;K,I,T,P:integer;
A:Array [1..5,1..3] Of Integer;
BEGIN Clrscr; Write('Файлдың атын енгіз: '); Readln(N);
Assign(X,N); Rest(X);
Write('Жаңа файлдың аты: '); Readln(New); Assign(X1,New);Rewrite(x1);
For K:=1 To 5 Do For I:=1 To 3 Do Readln(X,A[K,I]);
For k:=1 To 5 Do Begin
For P:=1 To 2 Do
For I:=P To 3 Do
If A[k,p]>A[K,I] Then Begin T:=A[K,P];A[K,p]:=A[K,I]:=T;End; End;
For K:=1 To 5 Do Begin
For I:=1 To 3 Do
Writeln (X1,A[K,I]); Writeln;End;
Writeln;Writeln('Файлдың соңы');Close(X);Close(X1); Readln End.

```

Текстік файлдың атын өзгерту.

```

PROGRAM Ft8;
VAR X: Text; N , New:String;
BEGIN Write ('Файлдың аты: '); Readln(N);
Assign(X,N);
Write('Файлдың жаңа аты: '); Readln(New);
Rename(X,New); Readln END.

```

Дискіден текстік файлды өшіру.

```

PROGRAM Ft9;
VAR X: Text; N:String;
BEGIN Write('Өшіретін файлдың аты: '); Readln(N);
Assign(X,N);
Erase(X); Readln END.

```

Типтері анықталған файлдармен жұмыс мысалдары.

```

PROGRAM FilJazy;
USES Crt;
TYPE Dosye=RECORD
A:Char; B:Integer; C,D,E:String; END;
VAR X:File Of Dosye; Y:ARRAY[1..5] Of Dosye; N:String; K:Integer;
BEGIN Clrscr; Write('Файлдың атын енгіз: '); Readln(N);
Assign(X,N);
Rewrite(X);
For K:=1 To 3 Do
With Y[K] Do Begin
Write('Қаласы: '); Readln(A);
Write('Машина нөмірі: ');Readln(B);
Write ('Сериясы: '); Readln(C);
Write(' Фамилиясы,аты: '); Readln(D);
Write(' Адресі: '); Readln(E);
Write(X,Y[k]); End; Close(X); Readln End.

```

Файлды оқу.

```
PROGRAM FilOku;
USES Crt;
TYPE Dosye=RECORD
    With Y[K] Do
    IF B=S Then Begin
        Writeln(K,'жазу: ');
Writeln('Қаласы: ',A); Writeln('Машина нөмірі: ',B);
Writeln('Сериясы : ',C); Writeln('Фамилиясы,аты: ',D);
Writeln('адресі: ',E); End; End;
Close(X); readln END.
```

ә) Адам аты бойынша іздеу:

```
PROGRAM Fillzdey2;
USES Crt;
TYPE Dosye=RECORD
    A:Char; B:Integer; C,D,E:String; End;
VAR X:File Of Dosye;
    Y:Array[1..15] Of Dosye; N:String;
    K:Integer; S:String;
BEGIN Clrscr; Write('Файлдың атын енгіз: '); Readln(N);
    Assign(X,N);
    Reset(X); Write('Адам атын енгіз: '); Readln(S);
    For k:=1 To Filesize(X) Do Begin Read(X,Y[K]);
        With Y[K] Do
        If D=S Then Begin
            Writeln(K,'жазу: ');
            Writeln('қаласы: ', A); Writeln('Машина нөмірі: ', B);
            Writeln('Сериясы: ',C); Writeln('Фамилиясы,аты: ', D);
            Writeln('Адресі: ', E); End; End;
        Close(X); Readln END.
```

Типтері анықталмаған файлдармен жұмыс мысалдары.

Бүтін сандардан тұратын файл жасау.

```
PROGRAM F1;
VAR X:File ; Y:Integer; K:Integer; N:String;
    BEGIN Write('Файлдың атын енгіз: ');
        Readln(N);
        Assign(X,N); Rewrite(X);
        For K:=1 To 10 Do Begin
            Read(Y); BlockWrite(X,Y,1) End;
        Close(X);
    Readln END.
```

Бүтін сандардан тұратын файлды оқу.

а) Бүтін сандардан тұратын типі анықталмаған файлды оқу :

```
PROGRAM F2;
```

```

VAR X:File; Y:Integer;K:Integer;N:Integer;
Begin Write('Файлдың атын енгіз: '); Readln(N);
Assign(X,N); Reset(X);
While Not Eof(X) Do Begin
BlockRead(X,Y,1); Writeln('Y=',Y) End; Close(X); Readln End.

```

Массив элементтерінен тұратын файл жасау.

```

PROGRAM F1;
VAR X:File; Y:Array[1..10] Of Integer;
    K:Integer; N:String;
BEGIN Write('Файлдың атын енгіз: '); Readln(N);
Assign(X,N); Rewrite(X);
For K:=1 To Do Begin
Read(Y[K]); BlockWrite (X,Y[K],1) End;
Close(X); Readln End.

```

Массив элементтерінен тұратын файлды оқу.

ә) файлды функцияны қолдану арқылы оқу

```

PROGRAM F2;
VAR X:File;
Y:Array[1..10] Of Integer;
    K:Integer; N:String;
Begin Write('Файлдың атын енгіз: '); Readln(N);
Assign(X,N); Reset(X); K:=0;
While Not Eof(X) Do Begin Inc(K);
BlockRead(X,Y[K],1);
Write('Y[' ,K, ']=', Y[K]); End;
Writeln; Writeln('Файлды оқу аяқталды '); Close(X);
Readln End.

```

Пайдаланылған әдебиеттер

Негізгі әдебиеттер:

1. Ж.Қ. Масанов, Б.А. Бельгибаев, А.С. Бижанова, Қ.Қ. Мақұлов. Алматы-2004. Turbo Pascal.
2. Б.Ж. Туркпенбаева, И.Н. Куценко. «Программирование на Turbo Pascal» Часть-I,II. Ақтау-2001 год.
3. К.З. Халықова, А.Р. Тұрғанбаева, Б.Ғ. Бостанов. «Программалау тілдерін оқыту. Delphi ортасы» Оқу құралы.-Алматы:Print-S.2005.
4. О. Камардинов. Информатика оқу құралы.Алматы-2008ж360бет.
5. «Секреты Delph на примерах» Орлик С.В. М.Ж Бином, 1996.,-136с.
6. Симонович С.В «Информатика» базовый курс
7. В. Фаронов . Объектно-рентированная среда Delphi Версия 6. Учебное пособие. «Русская редакция»,2002.
8. Борланд Р. Эффективная работа с Microsoft 7.0 Санкт-Петербург, изд. «Питер Пресс», 1996
9. Turbo Pascal ортасында программалау-II бөлім. Динисламов Ж. Қызылорда-2002ж.
10. В. Попов “Паскаль и Дельфи” Самоучитель, Питер 2004 ж.
11. Т. А. Павловская “Программирование на языке высокого уровня”

12. СПб Питер 2004 ж.
13. Фаронов В.В. "Turbo Pascal 7.0" Начальный курс. Учебно пособие
14. М. "Нолидж"-1998.
15. Камардинов О.Есептеуіш техника және программалау.-Алматы:
16. «Рауан», 97ж.
17. Грызлов В.И., Грызлова Т.П. "Турбо Паскаль 7.0" М, "ДМК"-1999.
18. Культин Н.Б. "Программирование в Турбо Паскаль и Delphi" Сант-
19. Петербург-1998.
20. Зуль Е.А. "Практическое программирование Turbo Pascal". М-1997.
21. Культин Н. "Турбо Паскаль в задачах и примерах". М-1999.
22. Семакин И.Г., Шестаков А.П. "Основы программирования".
23. М. "Мастерство"-2001.
24. .Малявко А.А. Теория формальных языков: Учеб. пособие: В 3 ч. Новосибирск:
Изд-во НГТУ, 2001. Ч. 1. 96 с.
25. .Малявко А.А. Теория формальных языков: Учеб. пособие: В 3 ч. Новосибирск: Изд-во
НГТУ, 2002. Ч. 2. 96 с.
26. Т.Пратт. Языки программирования. Разработка и реализация. М., Мир, 1979.

Қосымша әдебиеттер:

1. К.З. Халықова, А.Р. Тұрғанбаева, Б. Ғ. Бостанов. «Программалау тілдерін оқыту» Алматы-2005.