Ф

ФСО ПГУ 7.18.2/05

Қазақстан Республикасы Білім және ғылым министрлігі

С. Торайғыров атындағы Павлодар мемлекеттік университеті

Физика, математика және ақпараттық технологиялар факультеті

Информатика және ақпараттық жүйелер кафедрасы

"АЛГОРИТМДЕУ ЖӘНЕ ПРОГРАММАЛАУ ТІЛДЕРІ "

пәні бойынша зертханалық жұмыстарды орындауға арналған әдістемелік нұсқаулар

Павлодар

Ф

ФСО ПГУ 7.18.2/05

БЕКІТЕМІН
ФМ ж АТ факультетінің деканы
_____ Нурбекова Ж.К.
«____»_____2010ж.

Құрастырушы: Аға оқытушы Нұрғазина Б.Қ.

Информатика және ақпараттық жүйелер кафедрасы

5В070300 «Ақпараттық жүйелер» мамандығының студенттері үшін

"Алгоритмдеу және программалау тілдері"
пәні бойынша зертханалық жұмыстарға әдістемелік нұсқаулар

Кафедра мәжілісінде бекітілді, 200__ж. «___»_____ Хаттама №_____.

Кафедра меңгерушісі _____ А.Ж.Асаинова

Факультеттің әдістемелік кеңесінде құпталды,

200__ж. «___»_____ Хаттама №_____.

ӘК төрайымы _____ Ж.Г.Муканова

Зертханалық жұмыстардың мақсаты:

1. Студенттерге бағдарламалау тілінің негізгі құрылымдарын үйрету, командалар мен операторлардың қызметі мен пайдалану ережелерімен таныстыру.
2. Студенттердің алгоритмдік ойлау дағдыларын қалыптастыру.
3. Өз бетімен оқып-үйренуге, талдауға, бағдарлама бойынша нәтижені анықтауға үйрету.
4. Есепке алгоритмдер мен бағдарламалар құру, олардың тиімділерін таңдау және нәтижелерге талдау жасау әдістерін меңгерту.

4 тақырып Алгоритмдердің сызықтық құрылымын программалау.

№1 зертханалық жұмыс Меншіктеу операторы. Басқару операторлары. Мәліметтерді енгізу - шығаруды үйымдастыру.

Шығару, енгізу процедуралары

Мақсаты: Экранға әр түрлі хабарлама, нәтиже шығаруды және мәндерді енгізуді үйрену.

Теориялық бөлім

WRITE – айнымалының мәнін, өрнектің мәнін, текстi экранға шығарады.

Жазылу форматы**: write(*ln*) (<параметр>);**

Жазылу түрлері:

writeln (өрнек); – өрнектің мәнін экранға шығарады. writeln (sqr(25));

writeln(x); – х айнымалының мәні шығады.

writeln (‘текст’) – апострофтың ішіндегі текст шығады.

writeln (өрнек, ’текст’, айнымалылар);

writeln; – бос жол шығарады.

read(ln) – сандық мәндерді, символдарды, жолдарды клавиатурадан енгізеді.

Жазылу форматы: **read(*ln*)(айнымалылар тізімі);**

Жазылу түрлері:

readln(x) – х-тің мәні енгізіледі.

readln(a,b) – a,b мәндері бос орынмен ажыратылып енгізіледі.

readln- енгізу клавишасы басылғанын күтеді.

Read процедурасы орындалғанда, айнымалының мәнін енгізу үшін бағдарлама жұмысында кідіріс болады. Айнымалының мәнін оқушы клавиатурадан енгізеді. Егер read процедурасында айнымалылар тізімі көрсетілсе, онда тізімдегі айнымалылардың мәндері клавиатурадан *бос орынмен* ажыратылып енгізіледі. Әрбір read процедурасынан соң Enter клавишасы басылады.

1 жаттығу. Қадыр Мырзалиевтің «Ана тілі» тақпағының бір шумағын экранға шығару.

1. Бағдарлама тексін теріңіз

```
program text;
uses crt;
begin
clrscr;
```

```
Writeln;      Writeln('Ана тілі');      Writeln;
Writeln('Ана тілің арың бұл,');
Writeln('Ұятың боп тұр бетте.);
Writeln('Өзге тілдің бәрін біл,');
Writeln('Өз тіліңді құрметте');      Writeln;
Writeln('Қ.Мырзалиев');      Writeln;
readln;
  end.
```

Бағдарламаны орындап, нәтижесін көріңіз. Экранда орналасуына назар аударыңыз

2. Мына жолдарға өзгерістер енгізіңіздер

```
Writeln ('Ана тілі':20);
Writeln ('К. Мырзалиев':30);
```

Бағдарламаны орындап, нәтижесін көріңіз. Қандай айырмашылық бар? 20 және 30 цифрларын өзгертіңіз. Нәтижесіне түсініктеме беріңіз.

2 жаттығу. Жаңа терезеде мына жолдарды теріңіз

```
uses crt;
var x,y,a,b: byte;
begin
clrscr;          {1}
readln(x);       {2}
y:=10;           {3}
writeln(x);      {4}
writeln(x+y);    {5}
readln;
  end.
```

Орындап нәтижені қараңыз да, дәптерге жазып алыңыз

3. Мына {4} және {5} жолдарды өзгертіңіз:

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

Орындап нәтижені қараңыз. Қандай өзгеріс бар? Қорытынды жасаңыз.

4. Мына өзгерістерді енгізіңіз. {3} жолдан соң мына жолды теріңіз:

```
    a:= x*y;                                          {3a}
```

{4}, {5}жолдарды өзгертіңіз: `writeln (x,'*', y, '=' , a);` {4}

```
  Writeln ('x=', x:10, 'y=';10, 'a=',:10, a:10);  {5}
```

Орындап нәтижені қараңыз. Қандай өзгеріс бар? Қорытынды жасаңыз. {5}жолдағы **:** белгісі қандай қызмет атқарады?

3 жаттығу. Жаңа терезеде жазыңыз:

```
 uses crt;
var x,y,a,b:real; s:string;
begin
clrscr;                  {1}
x:=100; y:=3;            {2}
a:=x/y;
writeln ('a=';a);        {3}
```

```
writeln (‘a=’:10);
writeln (‘a=’:10:5);       {4}
writeln(‘a=’:10:2);
writeln(‘a=’:10:0);
writeln(‘a=’:0:5);        {5}
writeln(‘a=’:0:2);        {6}
writeln(‘a=’:0:0);
readln;
  end.
```

Бағдарламаны орындап, нәтижесін қараңыз. Экранға шығудың қандай айырмашылықтары бар? Санның бөлшек бөлігіндегі цифрлар санының өзгеруі неге байланысты?

4 жаттығу. Жаңа терезеде жазыңыз:

```
 uses crt;
var x,y,a,b:real; s:string;
begin
clrscr;                    {1}
readln(x,y);      a:=x/y;  {2}
writeln (‘x=’;x);          {3}
writeln (‘y=’,y);
writeln (‘a=’,a);          {4}
readln;    end.
```

Бағдарламаны орындаңыз. Экранға курсор шығады. Бос орынмен ажыратып екі сан енгізіңіз. Экранға шыққан х пен у-тің мәндері енгізілген сандарға ие болды ма?

Бағдарламаны бірнеше рет орындап, нәтижесін жазып алыңыз.

Сұрақ: Readln қандай қызмет атқарады? Ол орындалғанда экран қандай түрде болады? Курсордың орналасуы қалай?

5 жаттығу. Диалогтық программа құру

```
uses crt;
var s, a, b:string; x,y,z:byte;
begin
clrscr;                    {1}
Writeln(‘Сәлем!’);
Write(‘Атың кім?’); Readln(s); Writeln;
Write(‘Екі сан енгіз ’); readln(x,y);
Write(‘Бір  сан енгіз ’); readln(z);      Writeln;
Writeln(‘Үш  санның қосындысы = ’, x+y+z);      Writeln;
Writeln(‘Сау бол! ’,s); Writeln;
readln;    end.
```

Сұрақ: Write және Writeln айырмашылығы неде?

Өрнектерді бағдарламалау тілінде жазып есептеңіз.

1. $p = \frac{\sqrt{\lg 15 + 4{,}5} + ctg^3 3{,}1}{\cos 1{,}04 - 35}$ Жауабы: 402.1339

2. $k=\sqrt{\frac{e^{25-12,3}+121}{50\cdot 1,7}}-\frac{(458,78-48,97)\cdot 1,5}{25,78+\sqrt[3]{3}\cdot 35^4}$ Жауабы: 62.10672

3. $c=\left|125-6\cdot\frac{(0,75+3,25)^5+\sin^3 45^0}{tg1,2-\cos^2 3,5}\right|+100-2^7$ Жауабы: 3472.599

4. $d=\frac{|\cos 45^0-412|}{e^{1,8+7}+12}\cdot\sqrt{\frac{1525-418}{|1254-3564|+255,08}}$ Жауабы: 0.0406535

Пайдаланылатын әдебиет: [4], 5-24 беттер;

№2 зертханалық жұмыс. Шартты оператор. Логикалық өрнекті қолдану. Таңдау операторы. Шартты және таңдау операторларының қысқа және толық формалары.

Теориялық бөлім

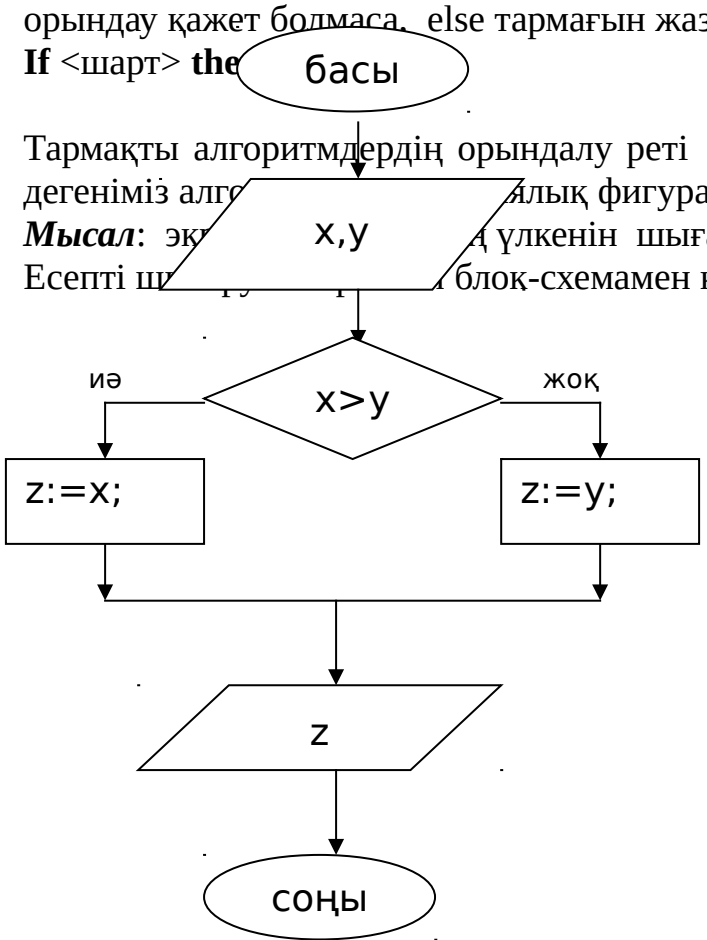
Белгілі бір шартты тексеру нәтижесіне байланысты екі түрлі іс-әрекеттің біреуі ғана орындалатын жағдайда шарттық оператор қолданылады. Шарттық оператордың жазылу форматы:

Толық түрі: **If** <шарт> **then** <1 оператор >
else <2 оператор >**;**

Шарттық оператор былай орындалады: Ең алдымен шарттағы логикалық өрнектің нәтижесі анықталады. Егер нәтиже ақиқат болса <оператор1> орындалады, ал нәтиже жалған болса <оператор2> орындалады.

Қысқаша түрі: Шарттық операторда шарт орындалмаған жағдайда еш нәрсе орындау қажет болмаса, else тармағын жазбауға болады:

If <шарт> **the**



Тармақты алгоритмдердің орындалу реті блок-схемада анық көрінеді. Блок-схема дегеніміз алг[illegible]ялық фигуралар арқылы бейнелеу.

Мысал: эк[illegible]ң үлкенін шығару керек.

Есепті ш[illegible] блок-схемамен көрнекі түрде көрсетуге болады.

```
program esep_1;
 var  x, y, z: real;              {x,y-берілген айнымалылар,  z-нәтиже}
 begin
     writeln (‘2 сан енгіз’);
      readln (x, y);              {бос орынмен  2 сан енгіземіз}
            if x>y then z:=x      {егер x >y болса, онда нәтиже x болады}
            else z:=y;            {әйтпесе нәтиже y болады }
      writeln (z);   readln;
 end.
```

Бағдарламаның орындалу барысында 5 және 7 сандарын енгіземіз. Айнымалы x-ке 5, y-ке 7 меншіктеледі (x:=5,y:=7). 5>7 шарты орындалмайды, нәтижесі жалған, сондықтан else –ден кейінгі оператор орындалады, ол оператор нәтижеге y-ті меншіктейді. Одан соң z-тің мәні экранға шығарылады. Экранға 7 жазылады.

Мысал: Енгізілген санның [-5;5] аралығында жататындығын анықтау.
Енгізетін санды x деп белгілейміз, типі – real. Егер x саны үшін x>-5 және x<5 шарттары бір уақытта орындалатын болса, онда x саны [-5;5] аралығына тиісті болады

```
program aralyk;
var    x: integer;
 begin
    writeln (‘x санын енгіз’);      readln (x);
     if (x>-5) and (x<5)
         then writeln (‘аралықта жатады’)
         else writeln (‘аралықта жатпайды’);
     readln;
end.
```

Тапсырма: Енгізілген сан теріс болса, оның таңбасын қарама-қарсыға ауыстыр. Есепті шешу үшін мына шарттық операторды қолдан. If x<0 then x:=-x;

Пайдаланылатын әдебиет: [2], 14-25 беттер; [1], 91-98 беттер;

Қабаттасқан шарттық оператор. Кейбір есептерді шешу кезінде, бірнеше

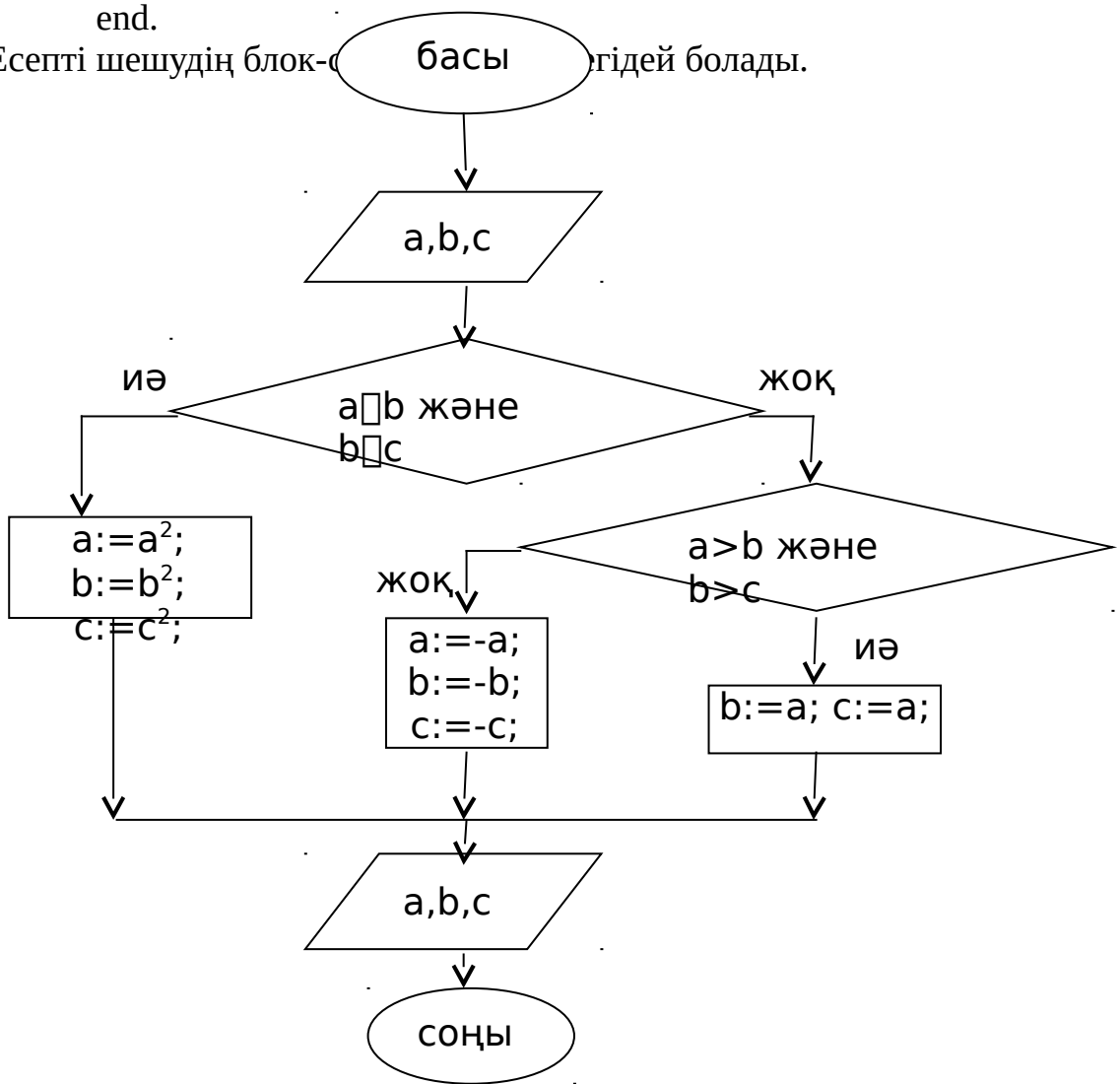
варианттарды қарастыруға тура келеді. Бұл жағдайда, бірнеше шарттық операторлар қолданылады, яғни then, else қызметші сөздерінен кейін, жаңа шарттық оператор жазылады.

Мысал: Бүтін a, b,c сандары берілген. Егер a□b□c болса, барлық сандарды өз квадратымен ауыстыр, егер a>b>c болса, әрбір санды үшеуінің ішіндегі ең үлкенімен, басқа жағдайда, әрбір санның таңбасын өзгерт.

Есептің берілгені бойынша: егер a□ b □c болса, онда a:=a^2, b:=b^2, c:=c^2;
егер a> b >c, онда c:=a, b:=a;
басқа жағдайда, a:=-a, b:=-b, c:=-c;

```
program esep_3;
var
   a , b , c : integer ;
begin
   writeln (‘a, b ,c сандарын енгіз’);      readln (a , b , c );
   if  (a<=b) and (b<=c)   then
                 begin
                    a:=sqr(a); b:=sqr(b); c:=sqr(c);
                 end
               else
                   if (a>b) and (b>c)   then
                        begin  c:=a; b:=a; end
                         else    begin a:=-a; b:=-b; c:=-c; end ;
    writeln (a:5,b:5,c:5);     readln;
  end.
```

Есепті шешудің блок-с гідей болады.



Тапсырма: Жоғарыдағы мысалдағы шарттық операторды былай өзгертсек:

```
 if  (a<=b) and (b<=c) then
begin
     a:=sqr(a); b:=sqr(b); c:=sqr(c);
     if  (a>b) and (b>c) then
        begin a:=c;  b:=c; end;
     else  begin a:=-a; b:=-b; c:=-c; end ;
 end.
```

бағдарламаның орындалуы қалай өзгереді? Есептің берілгені қалай болады. Блок-схемасын сыз.

№3-4 зертханалық жұмыс. Параметрлі цикл операторы. Көпмүшені есептеудегі арифметикалық алгоритмдер. Шарты алдын ала берілген және шарты соңынан берілген цикл операторлары. Кірістірілген циклдер.

Теориялық бөлім

Параметрлі қайталану. Қайталанатын әрекеттердің саны алдын-ала белгілі болғанда For операторы пайдаланылады. Бұл оператор ***параметрлі қайталану*** деп аталады, себебі, қайталану саны ***параметр*** немесе ***басқарушы айнымалы*** деп аталатын айнымалының мәніне байланысты болады. Бұл айнымалыда қайталану саны көрсетіледі. Қайталанатын әрекеттер ***цикл денесі*** деп аталады.

Жазылу форматы:

For <циклдің параметрі>**:=** <**s1**> **to** <**s2**> **do**
<оператор>; { өсу ретімен }

For <циклдің параметрі> **:=** < **s2**> **downto** < **s1**> **do** <оператор>;
{ кему ретімен}

Мұндағы: s1, s2 - параметрдің бастапқы және соңғы мәндері;

For … do - *циклдің тақырыбы*;
<оператор> –*цикл денесі*.

Цикл денесі жай немесе құрама оператор болуы мүмкін.

For операторы мынаны анықтайды:

– параметрдің өзгеру аралығын және цикл денесінің қайталану санын;
– параметр мәнінің өзгеруін (**to-өсу, downto-кему**);

FOR операторы параметр барлық мәндеріне ие болып біткенше, цикл денесінің

орындалуын қайталауды тоқтатпайды.
Егер **s2>s1** болса, цикл денесі **(s2-s1+1)** рет орындалады.
Егер **s1>s2** болса, цикл денесі орындалмайды.

Алғы шартты цикл. Егер іс-әрекеттің қайталану саны белгісіз, бірақ қайталану шарты белгілі болса, онда *while* немесе *repeat* операторлары қолданылады.

While (әзір) операторында қайталану шарты цикл денесінен бұрын тексеріледі. Сондықтан while операторы ***алғы шартты цикл*** *д*еп аталады.

Жазылу форматы:

While <қайталану шарты> **do**
<цикл денесі>

Қайталану шарты – бульдік (логикалық) өрнек, цикл денесі - жай немесе құрама оператор. Цикл денесі орындалудан бұрын шарттағы өрнектің мәні анықталады. Егер ол мән true болса, цикл денесі орындалады. Шарттық өрнектің мәні тағы да анықталады, егер нәтижесі false болса циклдың жұмысы аяқталып, while-ден кейінгі бірінші оператор орындалады.

Кейінгі шартты цикл. Циклдің бұл түрінде:
- қайталану шарты цикл денесінен кейін тексеріледі, сондықтан ***кейінгі шартты цикл*** деп аталады.
- қайталанатын іс-әрекеттер кем дегенде 1 рет орындалады.
- шарттың нәтижесі true болғанда циклдің орындалуы тоқтайды.

Жазылу форматы: **repeat**
<оператор;>
:
< оператор;> {цикл денесі}
until <циклді аяқтау шарты>;

Repeat және **until** сөздерінің арасындағы операторлар *цикл денесі* болады.

Кейінгі шартты циклдің орындалу тәртібі: алдымен цикл денесі орындалады, одан соң циклден шығу шарты тексеріледі. Егер нәтиже false болса, цикл денесі тағы да орындалады, егер true болса, циклден шығады.

1-мысал: 999 саны енгізілгенше бүтін сандарды енгізе отырып, олардың қосындысын табу.

```
…
x: integer;  sum: real;
begin
      sum:=0;
      repeat
            write (‘x-ті енгіз’);   readln(x);
             if x<> 999
                 then sum:=sum+x;
          until x=999;
   writeln (‘сандардың қосындысы=’,sum);   readln;
end.
```

Айнымалыларға түсініктеме:

x-енгізілетін бүтін сандар; sum-олардың қосындысы (real);

Бағдарламаның басында қосынды 0-ге теңестіріледі. Одан соң repeat сөзімен цикл ашылады. Цикл денесінде "х-ті енгіз" сұрауымен х айнымалының мәні енгізіледі. If операторы ол санның 999-ге тең емес екендігін тексереді. Егер тең болмаса, Sum қосындысының мәніне х саны қосылады. Циклдің соңындағы until x=999 циклдің аяқталу шартын тексереді. Егер х=999 болса, цикл аяқталады, until-ден кейінгі оператор орындалады. Ол оператор

writeln ('сандардың қосындысы=',Sum); нәтижені шығарады.

3-мысал: $\frac{1}{2}+\frac{2}{2\cdot3}+\frac{3}{3\cdot4}+\ldots+\frac{20}{20\cdot21}$ сан қатарының қосындысын табу.

Шешуі: Берілген есепте параметрі 1-ден 20-ға дейін 1 қадаммен өзгеретін (бөлшектің алымы) цикл құрған дұрыс. Бөлімінде алымы мен оған келесі санның көбейтіндісі жазылады. Егер алымын **i** деп белгілесек, бөлімі **i · (i+1)** түрінде болады.

```
program kosyndy;
var  i:byte; s : real;
begin
   s:=0;
    for i:=1  to 20 do
          s:=s+i/(i*(i+1));       { цикл денесі }
   write ('қосынды =',s);    writeln(y);    readln;
end.
```

4-мысал: 10-нан 99-ға дейінгі сандардың арасынан цифрларының қосындысы n-ге тең сандарды табу. (0<n<=18)

Шешуі: Мынадай айнымалыларды белгілейміз.

n- берілген сан; **p1**- санның оңдық разрядты цифрасы;

p2- санның бірлік разрядты цифрасы;

s- берілген санның цифрларының қосындысы;

k -ізделінді сан.

```
Program  san_kosyndy;
var k,n,p1,p2,s : integer;
begin
  writeln ('бүтін сан енгіз');    readln (n);   { бүтін санды  енгіземіз }
  for k:=10 to 99 do          { 10-нан 99-ға дейінгі сандарды қарастырамыз}
    begin                     { цикл денесінің басы }
      p1:=k div 10;
      p2:=k mod 10;          { бірінші, екінші цифраны  табамыз}
      s:=p1+p2;
      if  s=n  then writeln (k);          { егер қосынды n-ге тең болса,
    end;                                    берілген k санын шығарамыз }
        readln;
end.
```

Сұрақтар

1. Цикл дегеніміз не?
2. Циклді қолдануға мысалдар келтір.
3. Параметрлі қайталану қай кезде қолданылады?
4. Параметр деген не?
5. Параметрлі қайталанудың жазылу форматы.
6. For операторының орындалуын түсіндір.
7. Цикл денесі деген не?

ЗЖ 5-6-7. Ішкі программаларды ұйымдастыру. Ішкі программаларда параметрлерді тасымалдау Рекурсивті ішкі программаларды ұйымдастыру.

Процедуралар

Процедураның сипаттамасы ***процедураның таќырыбынан*** және ***денесінен*** т±рады.

Процедураның таќырыбы ***procedure*** деген резервтелген сµзден, процедураның атын білдіретін ***идентификатордан*** және жаќшаґа алынып, типтері кµрсетілген ***формальді параметрлер тізімінен*** т±рады. Процедура денесі бағдарламалыќ блоктан т±рады.

Процедураның жалпы т‰рі:

procedure <аты>[(формалдыќ параметрлер тізімі)];
    сипаттама бµлімі
    begin
        операторлар бµлімі
    end;

Процедура µздігінен орындалмайды. Ол аты бойынша негізгі бағдарламадан шаќырылады. Шаќыру жолында оның фактілік (наќты) параметрлері кµрсетіледі.

Фактілік және формальдыќ параметрлердің арасында мынадай сєйкестік болу керек:

● саны бірдей;
● типтері бірдей;
● жазылу реті бірдей;

Формальдыќ параметрлердің мынандай т‰рлері болады:

- параметр – мєндер;
- параметр – айнымалылар;

Параметр – мєндер. Параметр – мєндер негізгі бағдарламадан ішкі бағдарламаґа мєндерді беру ‰шін ґана ќолданылады, процедура нєтижесі негізгі бағдарламаґа ќайтарылмайды. Параметр – мєндер фактілік мєндерге ешќандай ыќпал жасамайды.

1-мысал: тµрт санды екі-екіден ж±птап, квадраттарының ќосындысын табу.

Б±л мысалда параметр – мєндер ќолданылады. Берілген сандар 2,5 пен 3,1; -7,2

жєне 5,3 болсын.

```
program  mander;
var k, z, x, y: real;
procedure sum_kv(a, b:real);          {a, b формальдық параметрлер}
begin                                 {процедура денесі}
    a:=a*a;    b:=b*b;
    writeln(’квадраттар ќосындысы=’, a+b);
end;
begin                                 {негізгі бағдарлама денесі}
    x:=2.5;  y:=3.1;
    sum_kv (x,y);
      z: = -7.2;   k:=5.3;
      sum_kv (z,k);
      readln;
end.
```

Бағдарламаґа т‰сініктеме

X, y, z, k - негізгі бағдарламадаґы фактілік параметрлер. Олар негізгі бағдарламада сипатталады.

Sum_Kv процедурасында екі санньњ квадраттыньњ ќосындысы экранґа шыґарылады; a мен b - процедурадаґы формальді параметрлер.

Процедура x, y параметрлерімен шаќырылґанда a=2.5 жєне b=3.1 мєндеріне ие болады да, олардыњ квадраттарыньњ ќосындысы есептеледі.

Экранґа мынандай хабарлама шыґады:

2.5 жєне 3.1 квадраттарыньњ ќосындысы

Бағдарлама ж±мысы процедурадан шыґып, негізгі бағдарламаньњ орындалуы жалґасады. z пен k-ньњ мєндері меншіктеледі. Процедура енді z, k параметрлерімен шаќырылады.

A=-7.2, b=5.3 мєндеріне ие болады. Енді осы екі санньњ квадраттарыньњ ќосындысы есептеледі.

Экранґа мынандай хабарлама шыґады :

-7.2 жєне 5.3 квадраттарыньњ ќосындысы

процедура ж±мысы аяќталып, негізгі бағдарламаґа оралады, негізгі бағдарлама ж±мысын аяќтайды.

Бағдарламаньњ орындалу тєртібі

1. Негізгі бағдарламадаґы айнымалылар сипатталады.
2. X, Y параметірлерімен Sum_Kv процедурасы шаќырылып, процедура орындалады, болґан соњ негізгі бағдарлама жалґасады.
3. Z, K параметрлерімен Sum_Kv процедурасы шаќырылып, процедура орындалады, болґан соң негізгі бағдарламаґа оралады.
4. Негізгі бағдарлама ж±мысын аяќтайды.

Параметр–айнымалы. Параметр – айнымалылар процедураньњ нєтижесін негізгі бағдарламаґа єкелу (ќайтару) ‰шін ќолданылады.

Параметр–айнымалылар негізгі бағдарламадағы фактілік параметрлерге ықпал етіп, оларды µзгерте алады.

2-мысал. Санның дєрежесін табуды процедура етіп алып, y=a4*x^4+a3*x^3+a2*x^2 мєнін есептеу бағдарламасын ќ±ру. М±ндағы, a4, a3, a2,x – клавиатурадан енгізіледі.

```
program  kosindi;
var x, a4, a3, a2, y, s,:real
procedure dareje (a: real; n: byte; var d: real);
    var i:byte;
    begin     {процедура денесі}
        d:=1;
        for i:=1 to n do
        d:= d * a
    end;
begin  {негізгі бағдарламаның  денесі}
    readln(x, a4, a3, a2);
    dareje(x,4,s); y:=s*a4;
    dareje(x,3,s);  y:= y +s*a3;   dareje(x,2,s); y:=y+s*a2;
    writeln (‘y=’, y); readln;
end.
```

Бағдарламадағы процедураға сипаттама. Процедура Dareje деп аталады. Онда a санының n дєрежесі есептеліп, нєтижесі d-ға меншіктеледі. A,n,d- формальдыќ параметрлер. Дєреженің нєтижесі d негізгі бағдарламаға ќайтарылатындыќтан var d:real сипаттамасы жазылады. і-формальдыќ параметр емес, сондыќтан ол процедураның сипаттама бµлімінде жазылған.

Процедура денесінде a санының n рет кµбейтіндісі есептеледі, a мен n-нің мєндері негізгі бағдарламадан беріледі.

Негізгі бағдарламаға сипаттама. Негізгі бағдарламада ќолданылатын фактілік параметрлер: x, a4, a3, a2, y, s; y-нєтиже, ал s-санның дєрежесінің нєтижесі.
x, a4, a3, a2-мєндері клавиатурадан енгізіледі.
x4, s-параметрлерімен dareje процедурасы шаќырылады.

Процедурадағы a=x, n=4 мєндеріне ие болып, x-тің 4 дєрежесі есептеледі. Нєтижесі S параметрімен негізгі бағдарламаға ќайтарылады, яѓни s:=d болады.
Негізгі бағдарламада y:=s*a4 мєні есептеледі.

x,3,s параметрлерімен процедура таѓы да шаќырылады, яѓни a=x, n=3 мєндерінде x-тің 3 дєрежесі есептеледі. Нєтижесі, яѓни d–ның мєні s-ке меншіктеледі, s жаңа мєнге ие болады да, негізгі бағдарламада y:=y+s*a3 есептеледі.
x, 2, s параметрлерімен процедура шаќырылып, іс-єрекет қайталанады.

Функциялар.
Теориялық бөлім

Паскаль тілінде пайдаланушы стандарттыќ функциялардан басќа, µз функциясын аныќтай алады. Ондай функциялардыњ сипаттамасы функцияныњ таќырыбынан жєне функция денесінен т±рады.

Функцияныњ жазылуыныњ жалпы т‰рі:

Function аты**(**формальді параметрлер тізімі): нєтиженіњ типі**;**
сипаттама бµлімі
begin
функция денесі
end;

Функция негізгі бағдарламадан аты бойынша шаќырылады. Функцияґа берілетін мєндер, ондаґы формальдыќ параметрге сєйкес болу керек. Функцияныњ таќырыбында функцияныњ аты жєне типтері кµрсетілген формальды параметрлер тізімі жазылады. Тізім жаќшаґа алынады. Жаќшаныњ сыртында функция нєтижесінің типі жазылады.

Функция денесінде кем дегенде бір меншіктеу операторы функцияныњ атына мєн меншіктейтін болу керек!

3-мысал. Санның дәрежесін табуды функция етіп алып, $z=(a^5+a^{-3}) / (2\cdot a^m)$ – өрнегінің мәнін есептеу.

```
program  z_funk;
var m: integer; a, z, r: real;
function dareje (n: integer; x : real): real;
    var i: integer; y: real;
    begin
       y:=1;
          for i:= 1 to n do
       y:= y*x; dareje:= y;
    end;
begin
  readln (a, m);
  z:= dareje (5,a);
z:= z+ dareje(3,1/a);
if m=0 then r:=1
  else if m>0 then r:= dareje (m,a)
            else r:=dareje (m,1/a);
 z:= z/(2*r);
writeln ('a=',a, 'm=':10, m, 'z=':10, z);
readln; end.
```

Бағдарламадағы функцияға сипаттама. Функцияның аты dareje деп аталады. Бұл функция x^n дәрежесін есептейді. N мен x- функцияның формальды параметрлері: n- дәреже көрсеткіші, типі integer; ал x n дәрежеге шығарылатын сан, типі real. Негізгі пограммаға қайтарылатын функцияның нәтижесі де нақты типті.

Функцияда формальды параметрлерден басқа, і және y айнымалылары сипатталған. і-қайталану санын білдіреді.

Y айнымалысы санды өз-өзіне і рет көбейтудің нәтижесін сақтайды. N-дәреже есептеліп болған соң, dareje функциясына y-тің мәні меншіктеледі.

Негізгі бағдарламаға сипаттама. Негізгі бағдарламада m,a,z,r айнымалылары сипатталады. m - дәреже көрсеткіші; a - берілген сан;
r - a^n дәрежесінің нәтижесі; z - нәтиже.

Бағдарлама орындалғанда a және m мәндері клавиатурадан енгізіледі. z:=dareje(5,a) жолында 5 және a фактілік параметрлерімен dareje функциясы шақырылады.

Функциядағы n жєне x параметрлері сєйкес мєндерге ие болады, яғни n:=5, x:=a. a^5 есептеліп, нєтижесі dareje айнымалысына меншіктеледі. Негізгі бағдарламаға оралған соң, функцияның мєні z айнымалысына меншіктеледі.

Z:=z+dareje(3,1/a) жолы орындалғанда функцияның формальды параметрлері мына мєндерге ие болады. n:=3, x:=1/a; себебі, $(a^{-n})=(1/a^n)$.

Функцияның орындалуы алдыңғыдай. Негізгі бағдарлама орындалған соң, z мєніне dareje мєні қосылады. Шарттық операторда n-нің мєні тексеріледі. Соның нєтижесіне байланысты r-ді анықтайды. Атап айтқанда, егер m=0 болса, онда r:=1;

егер m>0 болса, онда $r:=a^m$;
егер m<0 болса, онда $r:=(1/a^m)$;

m,a параметрлерімен dareje функциясы шақырылады, нєтижесі r-ге меншіктеледі. z:=z/(2*r) жолында z/(2*r) нєтижесі z-ке меншіктеледі.
Нєтиже экранға шығарылады.

Функциясы бар бағдарламаның орындалу тєртібі

1.Негізгі бағдарламадан фактілік параметрмен функция шақырылады.
2.Функциядағы формальді параметрлер фактілік мєндерге ие болады.
3.Функцияның нєтижесі анықталады, функцияның атына мєн меншіктеледі.
4.Негізгі бағдарламада функцияның мєні пайдаланылады.

Пайдаланылатын әдебиет: [2], 57-64 беттер; [1], 130-157 беттер;

№8-9 зертханалық жұмыс. Бір өлшемді массивті өңдеу.

Екі өлшемді массивті өңдеу.

Массив дегеніміз - бірдей типтегі элементтердің жиыны. Массив екі түрге бөлінеді:

- бір өлшемді (вектор)- бір ғана жолдан немесе бағанадан тұрады; A[45 12 -98 100]

- екі өлшемді (матрица)- жолдар мен бағаналардан тұрады;

$$A=\begin{bmatrix} 45 & -5 & 125 & 77 \\ 12 & 89 & 105 & -52 \\ 0 & 8 & -26 & 4 \end{bmatrix}$$

Массив элементтері массивтегі орналасу реті бойынша нөмірленеді, сондықтан массив реттелген тип деп аталады. Массив элементінің реттік номері индекс деп аталады. Индекстер тек скаляр типте болады. Индекстің типі массив мәндерінің шегін анықтайды.

Бір өлшемді массивтер. Сипатталу форматы:

<атау,...> : **array** [индекс типі] **of** [компонент типі];

Пайдалану мысалы: A: array[1..4] of integer – бүтін типтегі 50 элементтен тұратын А массивінің сипатталуы. Мұндағы, А-массивтің атын білдіретін атау; [1..4] - элементтердің бастапқы және соңғы индекстері;

Массивтің кез келген элементі массив атымен (атау) және индексімен анықталады. А[i] – А массивінің і-ші элементі. А[1]=45; А[3]=-98; Klas[16] – Klas массивінің 16-шы элементінің белгіленуі.

Егер тип типтерді сипаттау бөлімінде алдын-ала көрсетілсе, онда былай пайдаланылады:

```
type
      <тип аты> = array [индекс типі] of [компонент типі];
var
     <атау,...>: <тип аты>;
```

Пайдалану мысалы:
```
type   m1 = array [1..35] of real;
var   x,y: m1;
```

Бұл мысалда нақты 35 саннан тұратын m1 типі сипатталған. Ал х және у айнымалылары реттелген m1 типті, сондықтан олардың әрбіреуі нақты 35 саннан тұрады.

Массив сипаттамасында оның элементтер санын төрақты шамамен сипаттау ыңғайлы. Себебі, егер массив элементтерінің саны өзгеретін болса, онда ол Бағдарлама тексінің бәрінде емес, тек төрақтыны сипаттау бөлімінде ғана өзгертіледі.

Пайдалану мысалы:
```
const  n=20;
var  m: array[1...n] of real;
```

Массивтің әрбір элементі индекстелген, олар индекстелген айнымалылар деп аталады. Индекстелген айнымалылар да жай айнымалылар сияқты пайдаланылады.

D[3], Nom[17] жазулары D массивінің 3-элементін, Nom массивінің 17-элементін білдіреді. Егер массивті сипаттауда бір индекс болса, ол бір өлшемді (сызықтық таблица; вектор) деп аталады; екі индекс болса, екі өлшемді (тік төртбұрышты таблица; матрица) деп аталады. Klas:array [1.. 20,1.. 20] of integer; - Klas 20

жолдан және 20 бағанадан тұратын екі өлшемді массив.

Массивтермен жиі кездесетін әрекеттерді қарастырамыз.

1.Массивтерді тұтастай пайдалану. Массивті тұтастай қарастыру үшін, индекстері көрсетілмеген массив атауы пайдаланылады.

Егер A және B массивтері var a,b: array [1..20] of real; болса, онда оларға мынадай операцияларды қолдануға болады:

- салыстыру:

A=B, егер A массивінің элементі B массивінің сәйкес элементіне тең болса, нәтижесі true;

A<>B, егер A массивінің кем дегенде бір элементі B массивінің сәйкес элементіне тең емес болса, нәтижесі true;

- меншіктеу: A:=B; A массивінің әрбір элементіне B массиві-нің сәйкес элементі меншіктеледі. B - өзгеріссіз қалады.

2. Бастапқы мәндерін беру (толтыру, инициялизациялау):

- массивтің барлық элементтеріне базалық типке сай бір мәнді меншіктеу;

```
For i:=1 to 4 do            {A массивінің барлық  элементіне
A[i]:=0;                          де 0 мәні беріледі}
```

- массив элементтерінің мәнін клавиатурадан толтыру;

```
For i:=1 to 5 do          {C массивінің барлық 5 элементінің
Readln (C[i]) ;              мәндері клавиатурадан енгізіледі}
Readln D[5]               {D массивінің 5-ші элементін енгізеді}
```

- массивті кездейсоқ сандармен толтыру;

```
randomize;     for i:=1 to 7 do
k[i]:=random(105)
```

3. Массив элементтерін экранға шығару

- бағанамен;

```
for i:=1 to 4 do
    writeln (a[i]);
```

- бір жолға;

```
for i:=1 to 4 do

write (a[i]:5);  writeln;
```

4. Көшіру – бір массивтің әрбір элементін басқа массивтің сәйкес элементіне меншіктеу.

```
for i:=1 to 4 do
a[i]:=d[i];              {D массиві A массивіне көшіріледі}
```

5. Іздеу – массивтегі белгілі бір шартты қанағаттандыратын элементті іздеу.

Массивтегі нольге тең элементтерді санау мысалы:

```
k:=0;                    (0-ге тең элементтер k  айнымалысында саналады)
for  i:=1 to 4 do
    if a(i)=0 then k:=k+1;
```

6. Массив элементтерінің мәнін ауыстыру. Ол үшін типі массивтің типіне сәйкес келетін қосымша айнымалы қолданылады. Мысал: A(3;-2;4;7;0;8) массивінің бесінші және үшінші элементтерінің мәндерін ауыстыру керек.

Бағдарлама фрагменті мынадай болады:

```
v:=a[5];  a(5):=a(3];   a(3):=v      {v –қосымша айнымалы, a[5] мәнін  уақытша сақтайды}
```

Нәтижеде берілген массив мынадай болады: A (3; -2; 0; 7; 4; 8).

34 мысал. Бір өлшемді C массивінің элементтері индексінің квадратына тең болатын, ал K массивінің элементтері клавиатурадан енгізілетін және екі массивті экранға шығаратын бағдарлама құру:

Қолданылатын айнымалылар:

C(1..10), K(1..10) - массивтер.

i-циклдің параметрін және элементтің индексін білдіреді, бүтін типті.

Массивтің элементі оның индексінің квадратына тең болатын формуламен толтырылады, яғни C(i)=sqr(i). Массив элементінің мәні клавиатурадан енгізіледі, readln(K(i));

Массивте он элемент болғандықтан, а) және б) толтыру процестері циклде жүргізіледі. Массивтің мәндерін көру үшін, әрбір толтырудан соң оны экранға шығару керек.

```
program mas_tol;
var  c, к: array (1..10) of real;   i: integer;
begin
   for i:=1 to 10 do
       begin
          c(i) := sqr(i);            (массивті формуламен толтыру}
                 write ('массивтің ', i ,'-ші элементін енгіз');
                 readln (к(i));              {массивті клавиатурадан енгізу}
              end;
          writeln (' C  массиві ', '      K массиві  ');
           for i:=1 to 10 do  writeln (c(i), к(i):15 );
        end.
```

35 мысал. Массивті клавиатурадан толтырып, пайдаланушы енгізген санға тең элементтердің санын және ең бірінші тұрғанының номерін табатын бағдарлама көру.

Қолданылатын айнымалылар:
n – пайдаланушы енгізген ізделетін элементтің мәні;
a - массивтегі n-ге тең элементтердің номері;
b –массивтегі n-ге тең элементтердің саны;
i – циклдің параметрі және элементтің индексі;
k – массивтегі элементтер саны, ол бағдарламаның тұрақты шамаларды сипаттау бөлімінде жазылады. Себебі, егер элементтер саны өзгеретін болса, оны тұрақтыны сипаттауда өзгертсе жеткілікті.

```
program elem_iz;
const   k=10;
var      n:real;    a, b, i: byte;   m: array [1..10] of real;
begin
   for  i = 1 to k do         readln (m[i]);
      a :=0;                  { мұндай элемент жоқ деп есептейміз}
      b :=0;                  { басқа элемент табылған жоқ}
      write(‘массивтен  іздейтін элементті енгіз’);
      readln(n);
        for i := 1 to  do k
          if m[i] = n then    {массивтегі n-ге тең элементтерді іздеу}
          begin
              if  b = 0
               then a:= i;  {n-ге тең бірінші элементтің номерін табу}
                b := b+1;          { элементтердің санын бірге арттыру}
          end;
    if b=0
      then writeln (‘массивте мұндай элементтер жоқ‘)
      else
          begin
            writeln(n, ’-ге тең элементтің саны ‘ ,b);
            writeln(‘бірінші элементтің номері ‘,a);
          end;    end.
```

10 зертханалық жұмыс. Жолдық функциялар мен процедураларды пайдалану

Теорялық бөлім

Жол дегеніміз – ұзындығы 255-тен артпайтын символдар тізбегі. Тізбектің мағынасы болуы міндет емес. Мысал: ‘df56’, ‘*d-шар56‘, ‘*-4 лд’

Жолдар *string* жолдық типпен сипатталады. Жолдық типті анықтағанда ондағы символдар санын көрсетуге болады.

Жазылу форматы:

```
  type
```

<типтің аты>=string [символдар саны];

var <идентификатор>: <типтің аты>;

String типтегі айнымалыны типті алдын-ала сипаттамай-ақ көрсетуге болады: **var** <идентификатор>:string[жолдың ұзындығы];

N символдан тұратын жолға жадыдан N+1 байт бөлінеді. N байт-символдарды сақтау үшін, ал бір байт – жолдың ұзындығын сақтау үшін.

Жолдық өрнектер. Олар жолдық тұрақтылардан, айнымалылардан, функциялардан және операция таңбаларынан тұрады. Мысал:'ма'+'ма'

Жолдық процедуралар мен функциялар

Аты, жазылуы	**Қызметі**	**Ескерту**
1. **жою** delete(a,p,n)	a жолындағы p пози-циядан бастап, ұзындығы n символды жояды.	p<=255; нәтиже басқа айны-малыға меншіктелмейді.
2. **кіргізу** insert (a,s,p)	a жолын s жолына p по-зициядан бастап кіргізеді.	нәтиже басқа айны-малыға меншіктелмейді.
3. **типті ауыстыру** str(x,a)	x сандық шаманы жолға өңдеп, a-ға меншіктейді	x-ті шығару форматымен жазуға болады. X сандық типте, a –жолдық типте.
4. **типті ауыстыру** val(a,x,c)	a жолын сандық шамаға өңдеп, x айнымалыға орналастырады. A жо-лында бос символ болмау керек.	c-өңдеу нәтижесі, бүтін сан. Егер өңдеуде қате болмаса c=0 болады. A:string, x сандық типте, c:integer.
5.**ұзындық** length(a)	a жолының ұзындығын табады.	Нәтижені айнымалыға меншіктеуге болады.
6. **ретімен тіркестіру** concat(a,b,..s)	a,b,..s жолдарын сол ретімен тіркестіреді	-
7. **белгілеу** copy (a,p,n)	a жолынан p позициядан бастап, ұзындығы n символды белгілейді.	егер p>length(a) болса, нәтижесі бос символ; p>255 болса, қате.
8. **позиция** pos (a,s)	a жолы s жолында нешінші позицияда тұрғанын табады.	егер a жолы s жолында болмаса, нәтиже=0
9.**регистрді ауыстыру** upcase(ch)	кіші әріпті бас әріпке өзгертеді.	тек латын алфавитін ғана.

Мысалдар

Берілгені:	Қолданылған процедура, функция	Нәтиже

1.a:=’абвгде’	delete(a,4,2)	‘абве’
2.s1:=’интика’ s2:=’форма’	insert(s2,s1,3)	‘информатика’
1.var x:integer; a:string; ---------- x:=72584	str(x,a) str(-x:7,a)	‘72587’ ‘-72584’
2.var a:string; cod:integer; ---------- a:=’25’; a:=’14.2E+2’; a:=’14.2’	 val(a,x,cod) val(a,x,cod) val(a,x,cod)	 cod=0 cod=0 cod=5
3.st:=’1237’ st:=’klassio’	length(st) length(st)	4 7
4.a1:=’ ab’; a2:=’cd’; a3:=’ej’	concat(a1,a2,a3) concat(a1,’nm’,a2)	‘abcdej’ ‘abnmcd’
5.st:=’abcdefjk’	copy(st,2,4) copy(st,5,7)	‘bcde’ ‘efjk’
8.a1:=’abcdef’; a2:=’def’;	pos(a2,a1) pos(‘e’,a2) pos(‘k’,a1)	4 2 0
9.Ch:=’d’ A:=’x’	UpCase(Ch) UpCase(A) UpCase(‘a’)	‘D’ ‘X’ ‘A’

1-мысал. Енгізілген сөздегі ‘a’ әріптерін санау, ‘b’әрпіне ауыстыру.

1- әдіс: ***Алгоритм***:

1.Сөзді енгізу.

2.Сөздің бірінші әрпін белгілеу.

3.Белгіленген символды ’a’ символымен салыстыру.

4.Егер сәйкес болса, санауышты бірге арттыру, сол символдың орнына ’b’ символын қою;

5.Осылайша барлық символдарды қарастыру.

Бағдарлама.

```
program sanau;
var s:string;  n,l,i:byte;          {n-‘a’әрпін санауыш }
begin
   n:=0; readln(s); l:=length(s);    { l сөздің ұзындығы}
   for i:= 1 to l do
      if copy(s,i,1)=’a’             {кезектегі әріпті белгілеп,
           then                          ’a’ әрпімен салыстыру}
             begin
```

```
                delete (s,i,1); insert('b',s,i);
                n:=n+1;   { а әрпін санау}
                end;
        writeln ('сөзде,'n,' а әрпі бар');
    end.
```

2-әдіс. ***Алгоритм*:**

1.Сөзді енгізу.
2. 'а'әрпін санайтын санауыш енгізу.
3.Сөздегі 'а'әрпі тұрған позицияны тауып, сол орынға 'b' әрпін жазу.
4.Сөздегі барлық позициялар үшін 3 пунктті қайталау.
5.Нәтижені шығару.

Бағдарлама.

```
    program sanau;
    var
      s:string;  n:byte;
    begin
       write ('сөзді енгіз');  readln(s);    n:=0;
    while pos ('a',s)>0 do
        begin
            n:=n+1;   s[pos('a',s)]:='b';
         end;
    writeln ('сөзде',n,'  а әрпі бар');
    end.
```

№12 зертханалық жұмыс. Файлдармен жұмыс істеу процедуралары.

Теориялық бөлім

Өте үлкен көлемді ақпараттарды сыртқы жадыда сақтау ыңғайлы. Мысалы, оқу орнындағы студенттер туралы, кітапханадағы кітаптар туралы, т.с.с. мәліметтер. Бұл ақпараттар бағдарламада файлдар арқылы пайдаланылады.

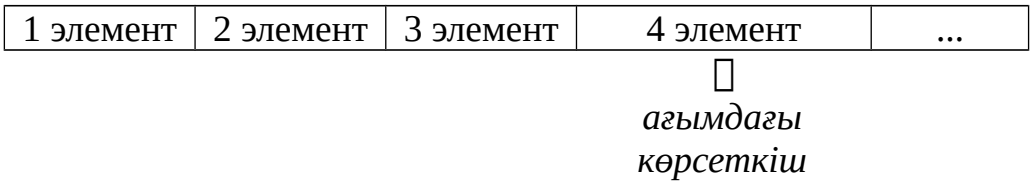
Файл дегеніміз сыртқы жадыда белгілі бір атпен сақталған деректердің жиыны.

Мәліметтерді файлдармен пайдаланудың себептері:

1. Бағдарламаның жұмыс барысында өте үлкен ақпаратты енгізу көп уақытты алады және адамды жалықтырады. Клавиатурадан енгізілген деректер мен экранға шығарылған нәтижелер сақталмайды, бағдарламаның жұмысы аяқталған соң жоғалып кетеді. Сондықтан, бұл ақпараттар алдын-ала дайындалып, дискіде сақталады да, қажетінше пайдаланыла беріледі.
2. Берілгендер файлын басқа бағдарламамен дайындап бірнеше бағдарламаны бір-бірімен байланыстыруға болады.
3. Бағдарламаның орындалу кезінде пайдаланушының қатысуы міндет емес.

Файлдық тип. Бір бағдарламада бірнеше файлмен жұмыс істеуге болады. Әрбір файл өз атымен аталады. Файлдағы компоненттер бір типте болады. Файлдың ұзындығы алдын-ала анықталмайды, ол құрылғының сыйымдылығына байланысты

болады. Файлды бір типтегі мәндердің шексіз тізімі деп қарастыруға болады. Файлдың элементтері нольден бастап нөмірленеді. Файлдың элементтері **ағымдағы көрсеткіш** арқылы көрсетіліп тұрады. **Ағымдағы көрсеткіш** бағдарлама жұмысына байланысты бір элементтен екіншіге ауысып тұрады. Кез келген уақытта файлдың бір элементіне ғана қол жеткізуге болады.



Файлдардың элементтерін *тізбекті* (последовательный) немесе *тікелей* (прямой) қарастыруға болады. Тізбекті файлдың элементтеріне *жазылу реті* бойынша қол жеткізіледі. Тікелей қол жетімді файлдың элементтеріне олардың *адресі* бойынша қол жеткізіледі. Сондықтан, *тікелей* файлдың кез келген элементін кез келген уақытта пайдалануға болады.

Дискідегі деректер файлы Паскаль бағдарламасымен файлдық айнымалы арқылы байланысады. Бағдарламада көпшілік жағдайда файлдық айнымалыны **f** арқылы белгілейді.

Бағдарламаның айнымалыны сипаттау бөлімінде файлдық айнымалы былай сипатталады.

var

файлдық айнымалы**: file of** элементтердің типі**;**

Мысал:

```
var
  f1, f2:file of integer;    f1, f2 - элементтері бүтін типтегі файл.
  s1 , s2:file of string;    s1, s2- элементтері жолдық  типтегі файл.
```

Файлдарды пайдалану әдістері.

1. Бағдарламаның басында файл мен файлдық айнымалыны байланыстыру қажет. Ол үшін мына процедура қолданылады:

assign(файлдық айнымалы, деректік файлдың аты);

2. Файлмен жұмыс істеу үшін алдымен оны ашу қажет. Файлды пайдалану мақсатына қарай ашудың екі түрі бар:

2.1. Файлдан деректерді оқу үшін –

Reset(файлдық айнымалы);

2.2. Жаңа файлды жасау үшін және оған деректерді жазу үшін

Rewrite(файлдық айнымалы);

3. Ашылған файлдың элементін оқу:

Read(файлдық айнымалы, айнымалы);

4. Ашылған файлға элемент жазу:

Write(файлдық айнымалы, айнымалы);

5. Файлмен жұмыс аяқталған соң файл жабылады:

Close(файлдық айнымалы);

1-мысал: Санды клавиатурадан енгізіп, san.txt файлына жаз.

```
program file_tip;
   var f:text;  s:integer;
```

```
  begin
    assign(f1, 'san.txt');      { san.txt файлы f1 айнымалысымен байланыстырылады. }
    rewrite(f1);                { f1 файлы деректерді жазу үшін ашылады}
    readln(s); write(f1, s); { s айнымалысының мәні  f1 файлына жазылады}
     close(f1);   { пайдаланылып болған соң, f1 файлы жабылады}
  readln;  end.
```

2-мысал: Деректерді san.txt файлынан оқып, экранға шығар.
Бағдарламаны жазудан бұрын san.txt файлы алдын-ала дайындалады.

```
program file_tip;
var f1: text;   s, n:integer;
begin
    assign(f1, 'san.txt'); { san.txt файлы f1 айнымалысымен байланысты}
    reset(f1);              { f1 файлы деректерді оқу  үшін ашылады}
    read(f1, s);            { f1 файлынан кезектегі элемент оқылып,  s айнымалысына
                            беріледі. }
    writeln(s);             {s айнымалысының мәні экранға шығарылады }
    close(f1);              {пайдаланылып болған соң, f1 файлы   жабылады}
    readln;
end.
```

№13-15 зертханалық жұмыс. Графиктік бейнелерді салу функциялары мен процедуралары.

Теориялық бөлім

Паскаль тілінде графикалық кескіндер жасау үшін, GRAPH модулі қолданылады. Бұл модульде 79 графикалық процедуралар, функциялар, тұрақты шамалар мен типтер орналасқан.

Графиканы жұмысқа қосу мына әрекеттерден тұрады:

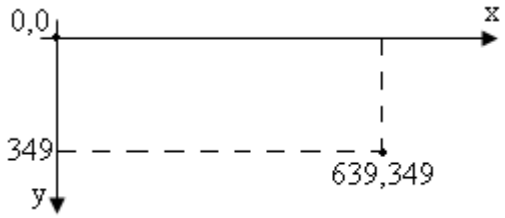
uses Graph; {Graph модулін, яғни графикалық процедуралар,
орналасқан кітапхананы іске қосу}
var dv, mv:integer; {dv, mv – екі айнымалысы графикалық
режимді іске қосады}
dv:=Detect; {detect мәнімен қажет графикалық драйвер
мен режим автоматты түрде іске қосылады}
InitGraph(dv,mv,'c:\tP7\BGI'); {graph модулінің
орналасқан жолы көрсетіледі}
If GraphResult<>grOk then Halt(1); {Графикалық режимді
іске қосудағы қатесі тексеріледі}

Осы әрекеттерден соң графикалық операторлар жазылады. Графикалық әрекеттер орындалып болған соң, графикалық режимді жабу керек.

6.**CloseGraph;** {Графикалық режимді жабу}.

Монитор экраны нүктелер жиынынан тұрады. Графикалық экранда координаталар

жазықтығының орналасуы төмендегідей:



Түстер таблицасы

Түстің номері	Түстің аты	Түстің номері	Түстің аты
0	Қара	8	Қою
1	Көк	9	Көгілдір
2	Жасыл	10	Ашық жасыл
3	Бирюза	11	Ашық бирюза
4	Қызыл	12	Қызғылт
5	Малина	13	Ашық малина
6	Қоңыр	14	Сары
7	Ашық сұр	15	Ақ

Графика элементі	**Паскальда жазылуы**	**Параметрлеріне сипаттама**
1	2	3
Нүкте	PutPixel(x,y,t)	x,y-нүктенің координатасы; t-түсі, санмен беріледі.
Кесінді	Line(x1,y1,x2,y2)	(x1,y1),(x2,y2) ұштарының координаталары
Сызық	LineTo(x,y)	x,y нүктелеріне дейін сызады
Тік төртбұрыш	Rectangle(x1,y1,x2,y2)	(x1,y1),(x2,y2)диагональдың координаталары.
Боялған төртбұрыш	Bar(x1,y1,x2,y2)	(x1,y1),(x2,y2)диагональдың координаталары.
Шеңбер	Circle(x,y,r)	x,y-центрдің координатасы; r-радиустың ұзындығы.
Эллипс	Ellipse(x,y,b,s,rx,ry)	b,s-эллипстік доғаның басы және соңы. b =0, s=360 эллипс салады. rx,ry – x және y бойынша радиустар.
Доға	Arc(x,y,b,s,r)	x,y,b,s,r- жоғарыда
Сектор	sector(x,y,b,s,rx,ry)	x,y, b,s, rx,ry - жоғарыда
Сызықтың түсі	SetColor(t)	t - түс номері, кестеде келтірілген.
Бояудың түсі	SetFillStyle(t1,t2)	t1 - бояу стилінің номері, t2-бояудың түсі.
Текст шығару	Outtext(‘текст’)	Текстi экранның сол жақ шетіне шығарады.
Текст	OutTextXY(x,y,’текст’)	Текстi көрсетілген координатаға шығарады

шығару		
Көпбұ-рыш салу	DrawPoly(n,pp)	n -нүктелер саны;
Тұйық ай-мақты бояу	FillPoly(t, pp)	t-бояудың түсі

Суретті бағдарламалау мысалы:

```
program suret;
uses graph;
var dv, mv:integer;
begin  dv:=detect;
    initgraph(dv, mv,'c:\tp7\bgi');
    if graphresult<>grok then halt(1);
    setcolor(5);
    rectangle(10,15,610,430);
    circle(350,100,50);
    setfillstyle(1,6);
    bar(100,250,200,300);
    readln;
    closegraph;
end.
```

```
program kopburish;
uses graph;
var pp:array[1..5] of PointType;
dv,mv,I,x1,y1:integer;
begin dv:=detect;
  initgraph(dv,mv,'c:\ tp7\bgi');
  pp[1].x:=300; pp[1].y:=50;
  pp[2].x:=400; pp[2].y:=50;
  pp[3].x:=350; pp[3].y:=150;
  pp[4].x:=150; pp[4].y:=200;
  pp[5]:=pp[1];
  DrawPoly(5,pp);FillPoly(3,pp);
  readln;
  closegraph;
end.
```

```
program kozgalys;
 uses Graph,CRT;
var
 x,y,i,dv, mv:integer;  x1,x2,y1,y2:integer;
begin
 dv:=detect;  initgraph(dv,mv, 'c:\bp\bgi');
 if graphresult <>grok then halt (1);
 x:=460;y:=400;            суреттің бастапқы координаталары
 for i:=1 to 200 do
 begin
    setcolor(6);
    circle(460,y,40);
    ellipse(460,y-40,0,180,20,20);
    ellipse(440,y-60,0,90,20,20);
    ellipse(480,y-60,90,180,20,20);                  сурет салынды
    ellipse(440,y-40,270,360,20,65);
    ellipse(480,y-40,180,270,20,65);
    circle(440,y-20,5); circle(480,y-20,5);
    circle(440,y+10,5); circle(480,y+10,5);
    delay(500);     пауза
     SetColor(0);
```

```
    circle(x,y,40);
    ellipse(460,y-40,0,180,20,20);
    ellipse(440,y-60,0,90,20,20);
    ellipse(480,y-60,90,180,20,20);                    сурет өшірілді,
   бұл
    ellipse(440,y-40,270,360,20,65);                  фрагментті
   cleardevice
    ellipse(480,y-40,180,270,20,65);                  процедурасымен
   circle(440,y-20,5);    circle(480,y-20,5);         алмастыруға
    circle(440,y+10,5);    circle(480,y+10,5);              болады.
     delay(500);
    y:=y-1;          жаңа координата
end;
 SetColor(6);
circle(x,y,40);
ellipse(460,y-40,0,180,20,20);   ellipse(440,y-60,0,90,20,20);
ellipse(480,y-60,90,180,20,20);  ellipse(440,y-40,270,360,20,65);
ellipse(480,y-40,180,270,20,65);
circle(440,y-20,5);       circle(480,y-20,5);
circle(440,y+10,5);       circle(480,y+10,5);
readln;
closegraph;
end.

program grafik;
 uses Graph,CRT;
var
 x,y,i,dv, mv:integer;   x1,x2,y1,y2:integer;
begin
 dv:=detect;   initgraph(dv, mv, 'c:\тр7\bgi');
 if graphresult <>grok then halt (1);
   line(300,50,300,400);    line(150,400,500,400);
   for i:=-100 to 100 do
   begin
      x:=i;  y:=trunc(sqr(x));
      putPixel(trunc(x/2+300),trunc(400-y/50),14);
    end;
readln;
closegraph;
end.
```

Пайдаланылатын әдебиет: [2], 101-104 беттер; [1], 336-408 беттер;

.