



Қазақстан Республикасының білім және ғылым министрлігі
С. Торайғыров атындағы Павлодар мемлекеттік университеті
Есептеу техникасы және бағдарламалау кафедрасы

«Бағдарламалау технологиясы» пәні бойынша
050704 «Есептеу техникасы және бағдарламалық қамтама»
мамандығы бойынша тәжірибелік жұмысын орындалуына

ӘДІСТЕМЕЛІК НҮСҚАУЛАР

№1 тәжірибелік шзі ые
Алгоритмдердің тармақты құрылымдарының бағдарламалау.
Шартты және шартсыз көшу операторлар

Жумыстын **максаты:** шартты оператор жұмысымен танысу және перехода. Операторының түр тандау операторы зерттеу. Бағдарламалардың құрастырушысы жанында олардың колдануға үйрену.

Теориялык мәлімдеме

1.1 If шартты операторы

Операторлардың қаралатын тобы бағдарламада тараулануды ұйымдастыруға рұқсат етеді. Тәуелділікте жиі, мысалы, қажетті ананың немесе басқа нәтижені біреуіннің іеке асыру немесе операторлардың басқа тобын (нұсқаулардың). СИ тілі үшін бұл мақсаттарда if операторлары колданылады (егер) - else (басқаша), switch (ауыстырып қосқыш) және goto (жүру).

If операторы түрді болады:

If (шарт тексеруі) нұсқау 1; else нұсқау 2;

Егер жақшаларда шарт нағыз мағынаны қабылдаса. нұсқау 1 орындалады. ал егер жалған - нұсқау 2 (1-суретті кара).

Мысалы:



```
If (a > b)
```

```
Z = a;
```

```
Else
```

```
Z = b;
```

Қажетті көңіл аудару нүктеге үтірлерден z кейін = a. Ол осында міндетті, if арналған сондықтан нұсқау тиісті еру, эркашан нүктелі үтірлерге аяқталады.

If операторында else сөзі жоқ бола алады. Мына оқиғада. егер жақшаларда шарт нағыз мағынаны қабылдаса. нұсқау 1 орындалады, ал егер жалған, онда нұсқау 1 өткізіледі. Және басқару бағдарлама мәтінімен келесі операторға беріледі.

Мысалы:

```
If (num >10) num=2* num;
```

```
Printf ("%d \n ", шип);
```

Эркашан шығару операторы орындалады, ал меншіктеу операторы томға тек қана оқиғада, егер нағыз шарт болса.

Көңіл болу операциялары, колданылатындар салыстыруға арналған, 1 кестеде келтірілген.

Кесте 1

Операция	Мәні
<	азырақ
<=	азырақ немесе бірдей
=	бірдей
>=	көбірек немесе бірдей
>	көбірек
	бірдей емес

Көңіл болу операциясын шатастыруға ермейді «= =» меншіктеу операциясымен «=».

Қарапайым бағдарламаны қарап шығамыз:

```
# include < stdio.h >
```

```
Main ()
```

```
Int a, b;
```

```
Puts ("a мағыналары және b енгізіңіздер");
```

```
Scanf ("%d %d " Да ,&b);
```

```
If (a > b) puts ("a көбірек b");
```

```
Else puts (" "); /* if- else оператор демонстрациясы
```

```
If (a = = b) puts (""); /* if оператор демонстрациясы else сөздері
```

```
}
```

Егер үшін орындалулар бағдарламалар енгізу сандар 5 және 3, анау экранда көрінеді

жол:

Ал көбірек **Б**

Кіріспеде сандардылардың 5 және 5. экранда көрінеді екі жолдар:

A b азырақ немесе бірдей

A b бірдей

Кейде шартты операторда if маңызды сөздерінен кейін немесе else if нұсқауын ереді. ішіне салынған оны атайды. If маңызды сөздерінен кейін және else сонымен қатар біреу нұсқау орнына нұсқауларды бірнешең мүмкін жазылған if. Оған else сөзі жақын алдыңғыға эркашан жатады, фигуралық жақшаларға тиісті болу.

Мысалы

```
If ( n > 0)
```

```
if ( a > b )
```

```
z = a;
```

```
else
```

```
z = b;
```

Else if ішкіге жатады. не және бос жерлерді аркасында көрсетілген. Егер басқа түсіндіру керек болса, қажетті тиісті бейнемен фигуралық жақшалар орын-орнына қою:

```
If ( n > 0)
```

```
{
```

```
if ( a > b )
```

```
z = a;}
```

```
else
```

```
z = b;
```

Else тап осы үлгісінде if сыртқыға жатады.

Күрделілерді контрукцияларда, if көп ұстаушылардың лоне else, if орынды ішіне салынғандар фигуралық жақшалармен көмкеру.

If - else ішіне салынған контрукциялары орнына шарт тексеруіне арналған күрделі айтылулар жиі мақсатқа лайықты қолдану. Қарапайым көңіл болулар бірнеше қисынды операциялар аркасында күрделі айтылулар біріктіру жолымен болып шығады. СИ тілінде қисынды операцияның үш болады, келтірілгендер кестеде 2.

Кесте 2

Операция	1 н
&&	және
	немесе
!	емес

Операция ! өте биік жетістікте болады. ол жоғарырақ, немен көбейтудің, және биік приоритет көбірек дөңгелек жақша тек қана болады. Операция приоритет! && жоғарырақ, немен операцияның ||, ал олар екі аласа приоритет көбірек болады, немен көңіл болу операциялары. Айтылу сондықтан

$$A > b \ \&\& \ b > c \ || \ b > d$$

Дәл осылай түсіндірілген волады:

$$f(a > b) \ \&\& \ (b > c) \ || \ (b > d).$$

Көбірек d. т. b е. c аралық арада және a немесе b орнында болады

Мысалы:

$$= \begin{cases} x^2 + 1 & x > 0 \\ x + 4 & x < 0 \end{cases}$$

Main ()

```
{ float x, y;
```

```
scanf("%f",&x);
```

```
if (x > 0)
```

```
Y = x*x+1;
```

```
else
```

```
Y = x + 4;
```

```
printf("%f %f", x, y);
```

```
}/
```

1.2 Шартсыз көшу операторы

Келесі түрде ұсынуға оны болады:

Goto таңба;

Таңба - мынау кез келген теңестіру.

Мысалы:

Goto a2;

Goto операторы көрсетеді. не бағдарлама орындалуы қажетті жалғастыру, нұскаудан бастай, алды қайсымен таңба жазылған. Бағдарламада жол міндетті тиісті болу, қайда таңба көрсетілген, нұсқау кос нүкте қойылған және жазылған, қайсыға асу тиісті орындалу.

Мысалы:

A2: k=5;

Керек емес бағдарламада таңбалар бейнелеп түсіндіру. Тілде сөзсіз асу оператор қолдануы СИ жағымсыз келеді. дәл осылай қалай ол бағдарлама құрылысты көрнектілігін бұзады.

1.3 Switch таңдау операторы

Switch операторы біреуінін таңдауға рұқсат етеді бірнеше балама, Ол келесі түрде жазылады:

```
Switch (айтылу)
```

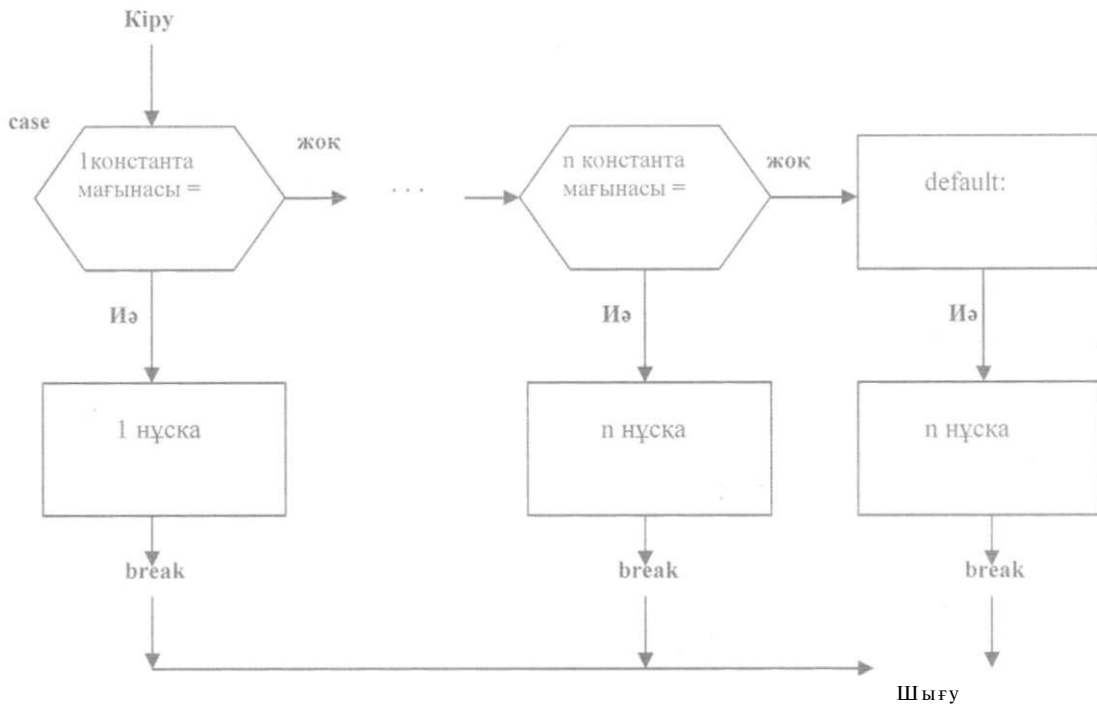
```
{ case тұрақты 1, түр 1; break;
```

```

Case n тұрақты, n түрі; break;
Default: n түр +1; break;
}

```

2-суретте көз алдына келген лайықты көркемдеу .



2-сурет

Switch операторында жақшаларда бүтін айтылу есептеледі (сұрыптаушымен оны атайды), және тұрақтылармен бәрімен оның мағынасы салыстырылады. Сәйкес келуде выполняется түр лайықты (бір немесе бірнешенің нұсқаулардың). Барлық тұрақтылар жазулар операторды тиісті болу әртүрлілермен. Түр default маңызды сөзімен (қалғанылар) іске асады, егер емес бір басқа келмеді (егер default сөзі жоқ болса, ал барлық салыстыру нәтижелері жағымсыз. анау бір түр емес орындалады. Келесі тексерулердің тоқтауына арналған табысты таңдауды - түрден кейін break оператор, switch, операторынан қамтамасыз етуші баяусыз шығу қолданылады.

Мысалы:

```
# include <stdio.h>
```

```
Main O
```

```
{ Char y;
```

```
Scanf("%c",&y);
```

```
Switch (y)
```

```
Case T:
```

```
Printf(" бұтақ 1\n ");
```

```
Break ;
```

```
Case '2':
```

```
Case '3':
```

```
Printf ("бұтақ 2 немесе 3\n");
```

```
Break ;
```

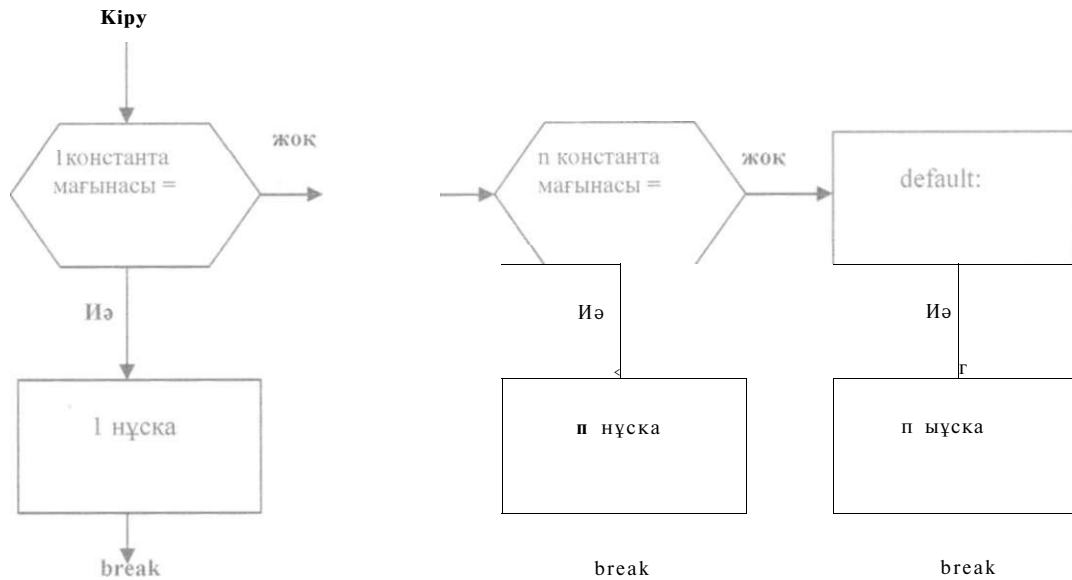
```
Default:
```

```
Printf("бұтақтың 1,2,3 жұмыс істемейді \n"); }
```

Case операторларының тұрақтыларымен бәрімен switch операторында оның мағынасы салыстырылады у. өзгергіш scant'операторы енгізеді егер 'Г. символ енгізу онда экранда жол

Case n тұрақты, n түрі; break;
 Default: n түр + 1; break;

z-суретте көз алдына келген лайықты көркемдеу .



Шығу >

2-сурет

Switch операторында жақшаларда бүтін айтылу есептеледі (сұрыптаушымен оны атайды), және тұрақтылармен бәрімен оның мағынасы салыстырылады. Сәйкес келуде выполняется түр лайықты (бір немесе бірнешенін нұсқаулардың). Барлық тұрақтылар жазулар операторды тиісті болу эртүрлілермен. Түр default маңызды сөзімен (калғанылар) іске асады, егер емес бір басқа келмеді (егер default сөзі жоқ болса, ал барлық салыстыру нәтижелері жағымсыз, анау бір түр емес орындалады. Келесі тексерулердің тоқтауына арналған табысты таңдауды - түрден кейін break оператор, switch, операторынан камтамасыз етуші баяусыз шығу қолданылады.

Мысалы:

```
# include <stdio.h>
```

```
Main O
```

```
{ Char y;
```

```
Scanf("%c",&y);
```

```
Switch (y)
```

```
Case 'Г:
```

```
Printf("бұтақ 1\n");
```

```
Break ;
```

```
Case '2':
```

```
Case '3':
```

```
Printf("бұтақ 2 немесе 3\n");
```

```
Break ;
```

```
Default:
```

```
Printf("бұтақтың 1,2,3 жұмыс істемейді \n"); }}
```

Case операторларының тұрақтыларымен бәрімен switch операторында оның мағынасы салыстырылады у. өзгергіш scanf операторы енгізеді егер 'Г. символ енгізу онда экранда жол

көршеді:

Бұтақ 1

Break оператормен switch ауыстырып қосқышынан шығу болады, және бағдарлама өз жұмысын аяқтайды. Егер енгізу символдар '2' немесе '3', анау экран толық шығарылған болады жол:

Бұтақ 2 немесе 3

Енгізуде - басқа символды басқару default маңызды сөзіне өтеді лоне экранда жол көрінеді:

Бұтақтың 1,2,3 жұмыс істемейді.

Тапсырма

Паскаль тілінде программасын құру:

- 1) Екі нақты сан берілген. Егер олардың бірі екіншісінен үлкен болса, экранға бірінші санды, басқа жағдайда экранға екеуін де шығару керек.
- 2) Екі нақты сан берілген. Егер бірінші сан екіншісінен артық болмаса, экранға оны нөлмен алмастырып шығару. басқа жағдайда сандарды өзгеріссіз экранға шығару керек.
- 3) Үш нақты сан берілген. (1,3) интервалында жататын санды экранға шығару керек.
- 4) X, y екі нақты сан берілген ($x \neq y$). Екеуінің кішісін олардың жарты қосындысымен, үлкенін екі еселенген көбейтіндісімен алмастырып. экранға шығару керек.
- 5) X, y нақты сандары берілген. Егер олар теріс болса. ер санды өзінің модулімен ауыстыру, егер біреуі ғана теріс болса, екі санды 0.5-ке арттыру, екі сан да теріс емес болса және бірде біреуі [0.5, 2.0] аралығында жатпаса. онда оларды 10-ға шегеру керек.
- 6) Үш нақты сандар берілген. Мәндері теріс еместерін квадраттау. мәндері ондарын кубтау керек.
- 7) Үш кез келген нақты сан берілген. Осы сандардың ішінен (2,6) интервалында жататын санды экранға шығару керек.
- 8) Үш нақты сандар берілген. Егер олардың біреуі қалған екеуінен үлкен болса. сол санды қалған екеуінің квадратымен. ал сандардың екеуі үшіншісінен үлкен болса, сол екеуін үшіншісінің квадрат түбірімен ауыстыру керек.
- 9) X, Y, Z 2 үш нақты санның ең үлкенін және ең кішісін табу керек.
- 10) X, Y нақты сандары берілген. Есептеу керек:
$$\begin{cases} |x-y|, & \text{егер } x > y \\ \sqrt{y-x+1}, & \text{болса жағдайда} \end{cases}$$

№2 тәжірнбелік жұмыс

Алгоритмдердің циклдік құрылымдарының бағдарламалау. Салынған цикл



Жұмыстың мақсаты: циклдік алгоритмдармен және операторлармен танысу. *ӘрШ* оператор қолдану ерекшеліктері игеру. Цикл барлық операторларының қолдануымен бағдарламалар біріктіру.

Теориялық мәлімдеме

1.1 While цикл операторы

Суреттеу:

While (айтылу) оператор;

Әрекет:

Соған дейін оператор орындалады, әзірше жақшаларда айтылу мағынасы нағыз. Айтылу мағына тексеруі оператор әрбір орындалуының алдында болады. Қашан айтылу мағынасы жалған. while циклы аяқталады. Егер айтылу жалған бастар, оператор бірге орындалмайды.

Түсінік:

Көріп калуға ереді. не кейін while маңызды сөздері және айтылудың. тұтқынды дөңгелек жақшаларға, нүкте үтірлерден қойылмайды.

Оператор цикл денесімен кейде аталады. Цикл денесінде әрекеттер тиісті орындалу. нәтижесінде қайсылардың басқарушы айтылу мағынасы алыстырады. Шексіз циклды болмаған жағдайда ала аламыз.

Үлгі:

```
/* while цикл демонстрациясы
```

```
# include < stdio.h >
```

```
Main O
```

```
hit i = 1
```

```
While ( getchar () != " ") i ++;
```

```
/* getchar операторы ( ) клавиатурадан кез келген символ енгізеді
```

```
Printf (" символы % ", i);
```

Символдардың жүйелілігінде енгізілген әріптері біріншінің келтірілген бағдарлама реттік нөмірді анықтауға рұқсат етеді. Ол while цикл қолдануын көрсетеді. денеде қайсыны барлығы бір нұсқау (i ++- бірлікке i бүтін сан мағына артуы). егер орындалуға мына бағдарламаны жіберу және символдардың жүйелілігі енгізу, мысалы:

```
AbFk ! Rgm, '
```

Онда экранда жол көрінеді:

6-шы таңба.

1.2 Do-while цикл операторы

Суреттеу:

Do while операторы (айтылу);

Әрекет:

Do-while операторында цикл денесі бір мәрте орындалады. Қалай дегенмен соған дейін цикл денесі орындалады. жақшаларда айтылу жалған мағынаны қабылдап жатады, Егер ол жалған циклға кіру жанында, анау оның денесі бір мәрте орындалады.

Түсінік: while сөзінен кейін және айтылудың, тұтқынды жақшаларға, үтірлерден нүкте

қойылады. Егер цикл денесінде көбірек біреу нұсқауыболса, онда цикл операторлары фигуралық жақшаларға болады.

```
Үлгі:  
/* do - while цикл демонстрациясы  
# include < stdio.h >  
Main ()  
f  
Int i =0; /* i =0, ал бірлікке емес  
Do i ++; while (getchar ()!="");  
Printf ("символы %". i):  
i
```

Бағдарлама. көз алдына келген жоғарырақ. сондай болады бағдарлама нәтиже do-while циклмен қазір жазылған ғой.

1.3 For цикл операторы

Суреттеу:

For (айтылу 1; айтылу 2; айтылу 3) оператор:

Әрекет:

Дөңгелектерді жақшаларда үш айтылу болады. Бірінші есепші инициализациясы үшін олардан қызмет етеді. Ол тек қана жүзеге асады бір мәрте - for циклы орындалуға бастағанда. Екінші айтылу шарт тексеруіне арналған қажетті. цикл денелері әрбір мүмкін орындалудың алдында жүзеге асады. Қашан жалған айтылу тұрады. цикл аяқталынады. Цикл денелері үшінші айтылу әрбір орындалу соңысында есептеледі. адымға сан жетілдіруі болады.

Түсінік: for циклы операторында нүкте үтірлерден жабылатын дөңгелек жақшадан кейін қойылмайды - үшеудің немесе барлық операторда үш айтылу жоқ бола алады, бірақ нүктенің олардың бөлушілер болмайды үтірлермен күсіру. Егер шексіз циклды 2. боламыз айтылу жоқ болса. Мысалы:

For (scanf("%d",&p)); p++) оператор;

СИ тілінде екі - операцияларды алдын ала ескерілген :(++)- бірлікке артуға арналған лоне (--)- бірлікке азаюға арналған операнды мағыналары. Операцияның ++ және — операндымен алды сияқты жазып алуға болады. дәл осылай және ол кейін. Бірінші оқиғада (++ п немесе — п) операнды мағынасы (п) лайықты айтылуда оның қолдануының алдында өзгереді. ал екіншіде (п ++ немесе п —) - оның қолдануынан кейін.

Егер жоқ болады айтылулар 1 және 3, цикл тұрады эквиваленттікпен while. Мысалы:

For (; a <20;) оператор:

Әрбір айтылулардан түзеле алады бірнеше айтылу, операциямен біріккендердің «үтір». Мысалы:

```
For (i =0, j =1; i<100: i ++.j ++) a [i] = b [j];
```

Цикл денесі фигуралық жақшаларға болады. егер оған көбірек бір оператордың.

Үлгі:

```
/* for цикл демонстрациясы  
# include < stdio.h >  
Main ()  
{int i, j =1, k ;  
For (i =1; i<=3; i ++)  
Printf (" Минскі \t");  
/* for циклында үш рет штығару функциясы орындалады.  
/* осында i - басқарушы өзгергіш циклдың  
Printf ("\ цикл қайталауларының п санын көрсетіңіз \п");
```

```

scanf("%d",& k);
for (i =1; i<= k ; i ++ )
    {j;
    printf("% d " J);
    } /* екі нұсқау осында (көбірек біреудің), олар сондықтан фигуралық жакшаларға
болады
J = i;
printf("\n ");
/* 1 өзгергіш j мағынаны меншіктеледі және мензер аудармасы жүзеге асады
/* for келесі циклында ана әрекеттері орындалады, және алдындағыда
for (i=1; i<= k : i ++ ) printf("% d " j i);
i

Нәтижелерді келесі бағдарлама орындалулары:
Минскі Минскі Минскі
Цикл қайталауларының санын көрсетіңіздер
5
12624120
12624120

```

1.4 Break операторы

Суреттеу:

Break қате табылуы артынан цикл орындалулары тоқтау үшін қолданылады, цикл тақырыбысында шартқа қосымша ұйымына арналған, шексіз цикл тоқтауына арналған.

Үлгі:

```

while (st >0&& st <25)
    {
    if(st ==4|| st ==8|| st ==12)
        break;
    }

```

Цикл жұмысы толық токталады. қандай тек қана нағыз if операторында шарт тұрады.

1.5 Continue операторы

Суреттеу: continue

Әрекет:

Мынау оператор қолданыла алады циклдардың үшеу үлгі барлықтарының. Қалай және break операторы оқиғасында. ол бағдарлама орындалулары мінез-құлық өзгертуіне ертіп әкеледі. Бірақ аяқтау орнына цикл жұмыстары continue оператор барысы кіргізу деп атайды "қалғанның" итерация бөлімдері және келесі басқа асу.

Үлгі. Continue break оператор алдындағы бағдарламада ауыстыруға болатын

```

while (st >0&& st <25)

```

```

if(st ==4|| st ==8! st ==12)

```

```

continue; ;

```

If операторында нағыз шарт жанында continue операторы жүргенге кіргізу деп атайды үшін оларға операторлардың цикл денелері және келесі итерация басына асу жүзеге асады.'

Тапсырма

- 1 Берілген 10-нан 99 дейін сан аралығынан, цифрлар қосындысы N санына тең сандарды шығару. ($0 < N \leq 18$).
- 2 Цифрларының қосындысы, берілген N санына тең, үш орынды сандардың санын есептеу.
- 3 (Мысалы N -ді 3 деп алсақ, онда цифрларының қосындысы 3-ке тең: 102, 111, 120, 201, 210, 300 сандар екен, яғни ондай сан 6 болтаны).
- 4 25 тең 125 дейін сандардың кубтарының қосындысын есептеу.
- 5 Екі орынды сандар арасынан, цифрлар квадраттарының қосындысы 13-ке қалдықсыз бөлінетіндерін табу.
- 6 Екі орынды сандар арасынан мына қасиетке сай, іздеу программасын құру: егер, санның цифрлар қосындысына, осы қосындының квадратын қосқанда, санның өзі шығу керек.
- 7 Кейбір үш орынды сандардың, квадраттарының соңғы үш цифрі санның өзіне тең. Сондай сандарды іздеу программасын құру.
- 8 Төрт орынды сандар арасынан, 133 бөлгенде, қалдығы 125, ал 134-ке бөлгенде қалдығы 111 болатын сандарды іздеу программасын құру.
- 9 100-ден кіші, оң тақ сандардың қосындысын есептеу.
- 10 A дан B -ға дейін аралығында, 4-ке еселі, бүтін он сандар қосындысының табу керек (A мен B -н мәндері клавиатурадан енгізіледі).