

4 ТАРМАҚТАЛУ АЛГОРИТМІ

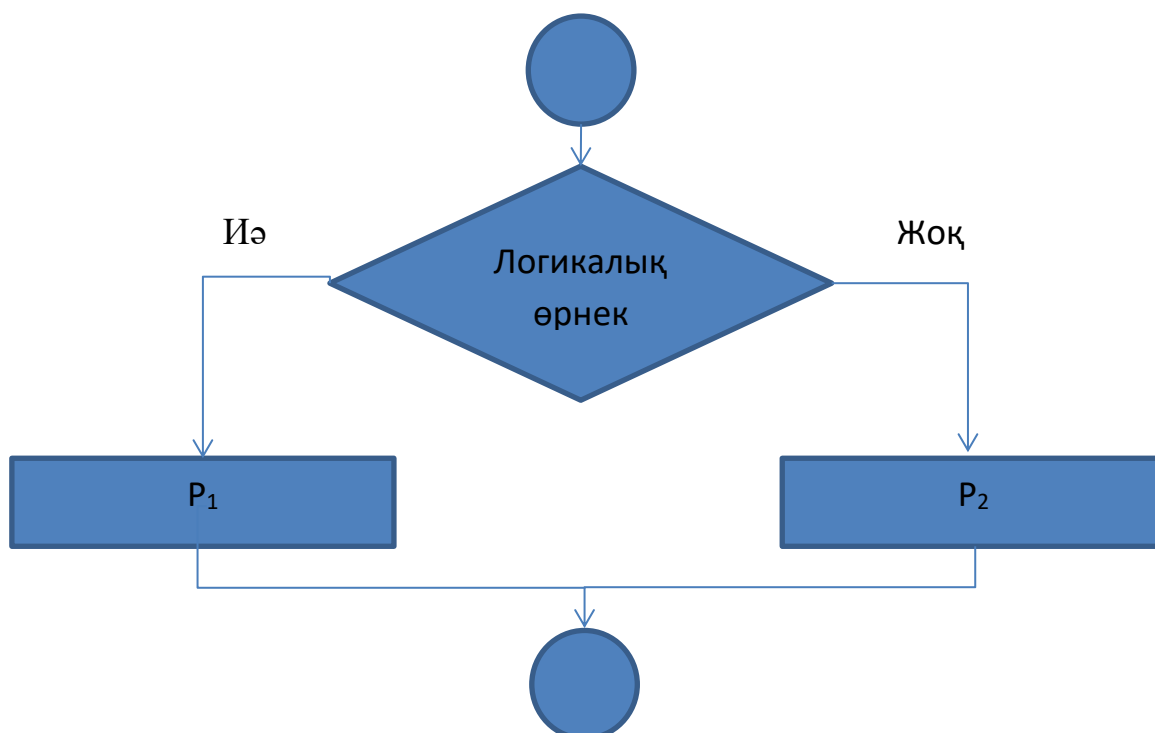
Бұрын біз деректерді енгізу және мәндерді айнымалыларға қалай тағайындау керектігін қарастырдық. Енді бағдарлама барысында қалыптасқан жағдайға байланысты қосымшаны орындаудың әртүрлі ағындарын ұйымдастыру мәселесіне тоқталайық. Ол үшін **тармақталу** алгоритмі дегеніміз не?

Алгоритм қадамдарының реттілігі белгілі бір шарттардың орындалуына байланысты өзгерсе, алгоритм **тармақталу** деп аталады. Шарт дегеніміз - екі мәннің біреуін қабылдай алатын логикалық өрнек: "иә" - егер шарт дұрыс болса (ақиқат), ал "жоқ" - егер шарт дұрыс болмаса (жалған).

Тармақталған алгоритмді бағдарламаларда қарапайым, қысқартылған, құрама операторлардың көмегімен, сондай-ақ көп мағыналы тармақтардың құрылымымен жүзеге асыруға болады. Төменде осы нұсқаларды толығырақ қарастырыңыз.

4.1 Қарапайым шартты оператор

Қарапайым шартты оператор құрылымының алгоритмінің жалпы түрі 30-ші суретте көрсетілген.



Сурет 30 – Шартты оператордың блок-схемасы

Қарапайым шартты оператордың синтаксисі келесідей :

if Логикалық өрнек:

```
P1  
else:  
P2
```

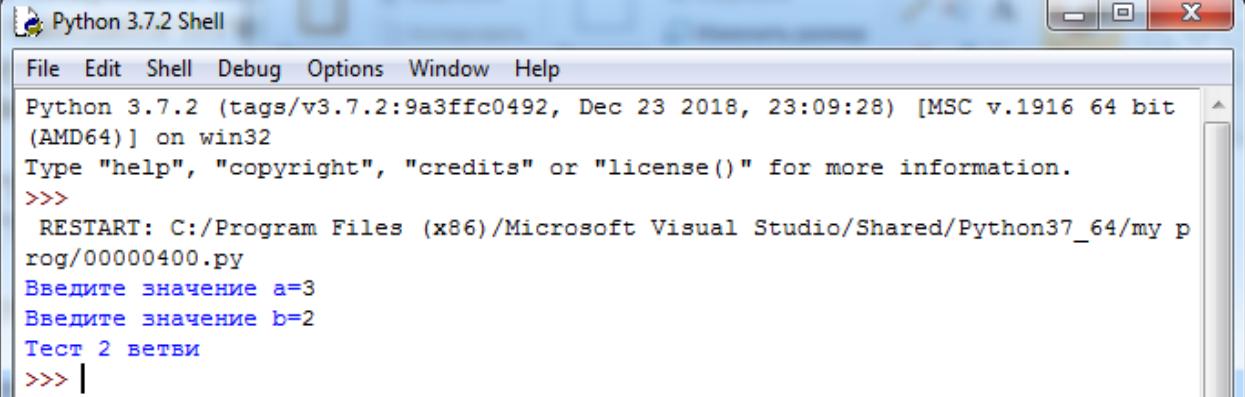
мұнда **if** (егер), **else** (әйтпесе) - сақталған сөздер, а P₁, P₂ - операторлар.

Қарапайым шартты оператор келесі алгоритм бойынша жұмыс істейді: алдымен логикалық өрнек есептеледі. Егер нәтиже **true** (ақиқат) болса, онда P₁ операторы орындалады, ал P₂ операторы өткізіп жіберіледі. Егер нәтиже **false**(жалған) болса онда P₂ операторы орындалады, ал P₁ операторы жіберілмейді.

Мысалы, келесі кодта екі айнымалыны салыстыру нәтижесіне байланысты бір немесе басқа жауап көрсетіледі. If операторы-блок операторы, сондықтан кодтағы шегіністер міндетті болып табылады. Операторлар блогына кіру қос нүкте арқылы жүзеге асырылады:

```
a=int(input("Введите значение a="))  
b=int(input("Введите значение b="))  
if a<b:  
    print("Тест 1 ветви")  
else:  
    print("Тест 2 ветви")
```

31-ші суретте **a=3**, **b=2** мәндер үшін қарапайым шартты операторды орындау мысалының скриншоты көрсетілген



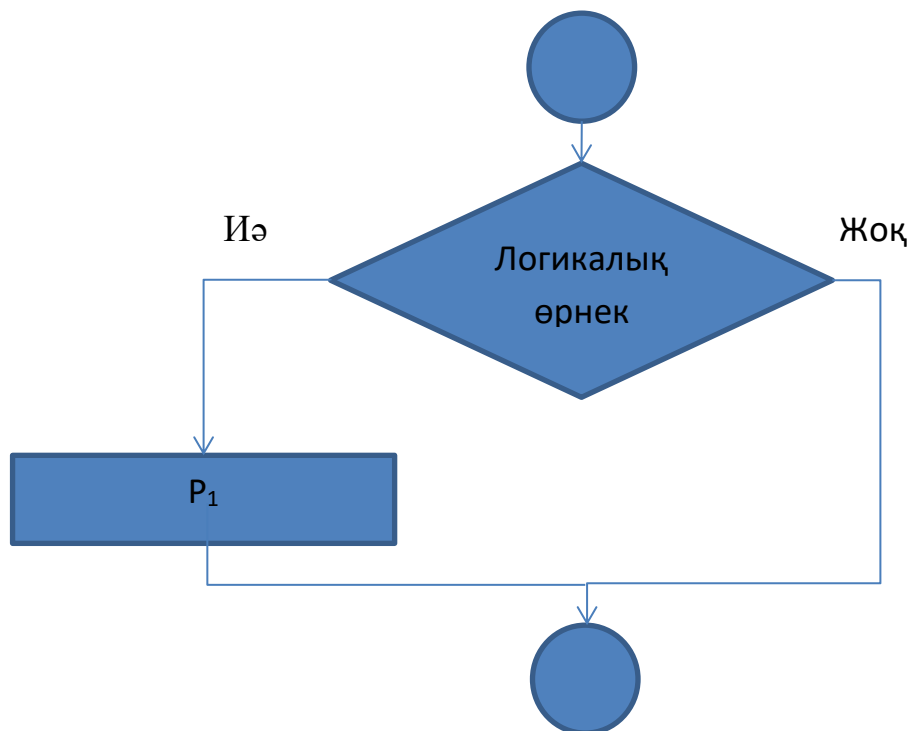
```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit  
(AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p  
rog/00000400.py  
Введите значение a=3  
Введите значение b=2  
Тест 2 ветви  
>>> |
```

Сурет 31 – a=3, b=2 мәндері үшін қарапайым шартты операторды орындалуы

4.2 Қысқартылған шартты оператор

Егер белгілі бір әрекетті тек тексерілген **шарттың** шындығымен орындау қажет болса, онда бұл жағдайда **қысқартылған шартты оператор** қолданылады. *Қысқартылған шартты оператор* құрылымының

алгоритмінің жалпы түрі 32-ші суретте көрсетілген.



Сурет 32 – Қысқартылған шартты оператордың блок-схемасы

Қысқартылған шартты оператордың синтаксисы:

if Логикалық өрнек:
P1

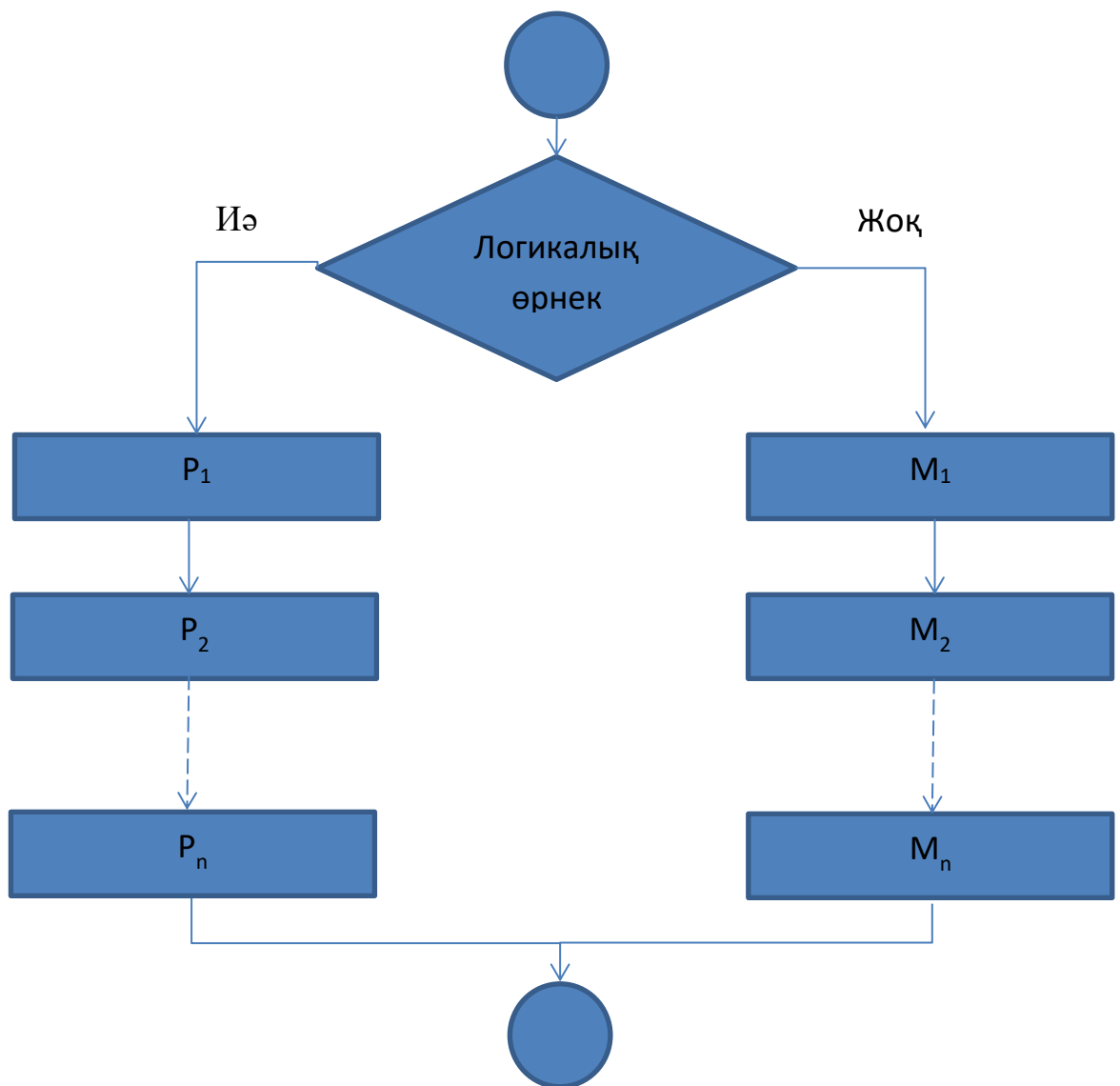
мұнда **if** (егер) – сақталған сөз, ал P₁ - оператор.

Мысалы, төмендегі листингте логикалық өрнектің ақиқатымен сәйкес хабарлама шығады "Тест 1 ветви»:

```
a=int(input("Введите значение a="))
b=int(input("Введите значение b="))
if a<b:
    print("Тест 1 ветви")
```

4.3 Құрамдас шартты оператор

Егер белгілі бір жағдайда операторлардың белгілі бір тізбегін орындау қажет болса, онда олар бір құрама операторға біріктіріледі. Құрама шартты оператор құрылысының алгоритміндегі жалпы түрі 33-ші суретте көрсетілген.



Сурет 33 – Құрамдас шартты оператордың блок-схемасы

Құрамдас шартты оператордың синтаксисы келесі түрде:

if Логикалық өрнек:

P_1
 P_2
 \cdot
 \cdot
 P_n
else:
 M_1
 M_2
 \cdot
 \cdot
 M_n

мұндағы **if**, **else** – сақталған сөздер, ал $P_1, P_2, \dots, P_n, M_1, M_2, \dots, M_n$ -

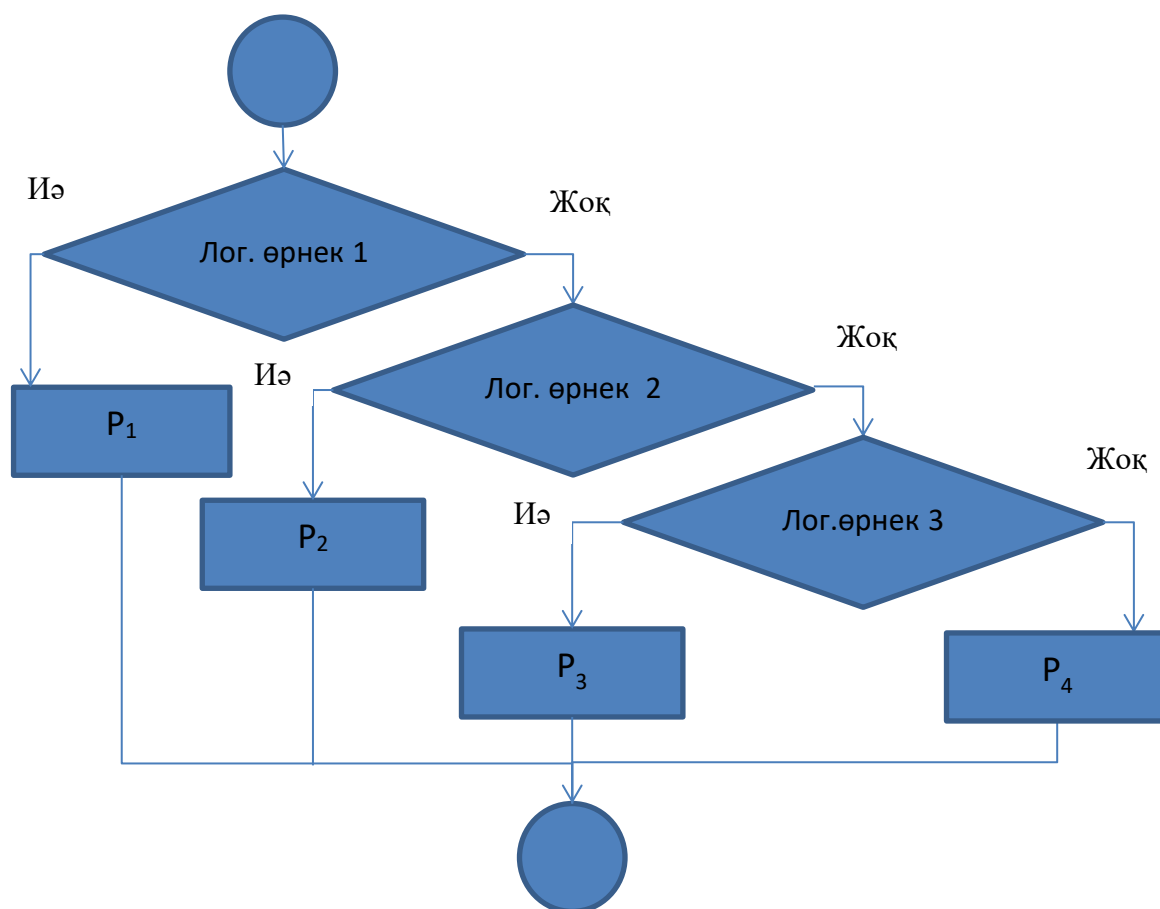
операторлар.

Мысалы, төмендегі листингте шартты оператордың әр тармағында біз екі операторды орындағымыз келеді. Содан кейін олардың әрқайсысын жолдың басынан бастап төрт бос орынға оңға жылжыту керек:

```
a=int(input("Введите значение a="))
b=int(input("Введите значение b="))
if a<b:
    a=a+b
    print("Сумма двух чисел a+b=", a)
else:
    a=a*b
    print("Произведение двух чисел a*b=", a)
```

4.4 Көп мағыналы тармақтар

Мәселені шешудің жолын екіден емес, бірнеше мүмкіндіктен таңдау өте жиі кездеседі. Бағдарламалауда бұл қадамды бірнеше шартты операторлардың көмегімен жүзеге асыруға болады. Көп мәнді тармақтарды құру алгоритміндегі жалпы көрінісі 34-ші суретте көрсетілген.



Сурет 34-Көп мағыналы бұтақтардың блок-схемасы

Көп мағыналы бұтақтардың синтаксисы келесі түрде:

if Логикалық өрнек 1:

P_1

elif Логикалық өрнек 2:

P_2

elif Логикалық өрнек 3:

P_3

else:

P_4

Мұнда **if, elif, else** – сақталған сөздер, аа P_1, P_2, P_3, P_4 - операторлар.

Бұл құрылымның жұмыс істеу алгоритмі келесіде. Егер **логикалық өрнек 1** ақиқат болса, онда осы тармақтағы оператор немесе оператор блогы орындалады, әйтпесе бұл оператор немесе блок өткізіліп жіберіледі. Егер **if** операторынан кейінгі логикалық өрнек жалған болса, онда **elif** операторынан кейінгі **логикалық өрнек 2** талданады. Егер ол шын болса, онда осы тармақтағы оператор немесе оператор блогы орындалады, әйтпесе бұл оператор немесе блок өткізіліп жіберіледі. Соңғы **else**-ден кейінгі мәлімдемелер алдыңғы барлық логикалық өрнектер жалған болған жағдайда ғана орындалады. Мұндай құрылымдағы **if** шартты операторлары **кірістірілген** деп аталады.

Мысалы, Листингте бағдарламаның үш тармағын тестілеу процесі көрсетілген. Пайдаланушы **a** және **b** ұяшықтарындағы бастапқы деректерді өзгерте алады және әр уақытта белгілі бір нәтиже алады.

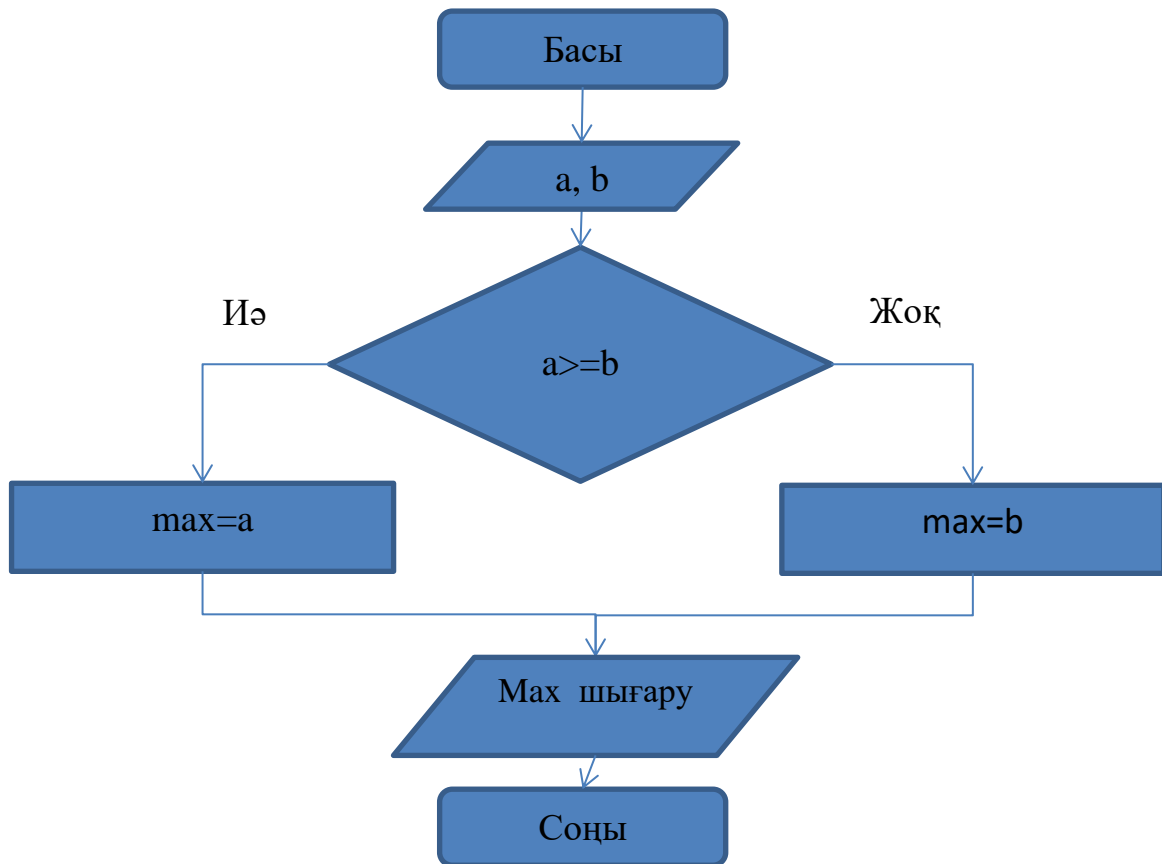
```
a=int(input("Введите значение a="))
b=int(input("Введите значение b="))
if a<b:
    a=a+b
    print("Сумма двух чисел a+b=", a)
elif a==b:
    a=a*b
    print("Произведение двух чисел a*b=", a)
else:
    a=a-b
    print("Разность двух чисел a-b=", a)
```

4.5 Максималды және минималды элементтерді іздеу алгоритмдері

Python бағдарламалау тілін одан әрі үйренуде қажет болатын максималды және минималды мәндерді табу алгоритмдерін қарастырыңыз

Есеп 4.1. Екі санның арасындағы ең үлкенін табыңыз.

Шешімі. Есептің шешу алгоритмы 35-ші суретте көрсетілген.



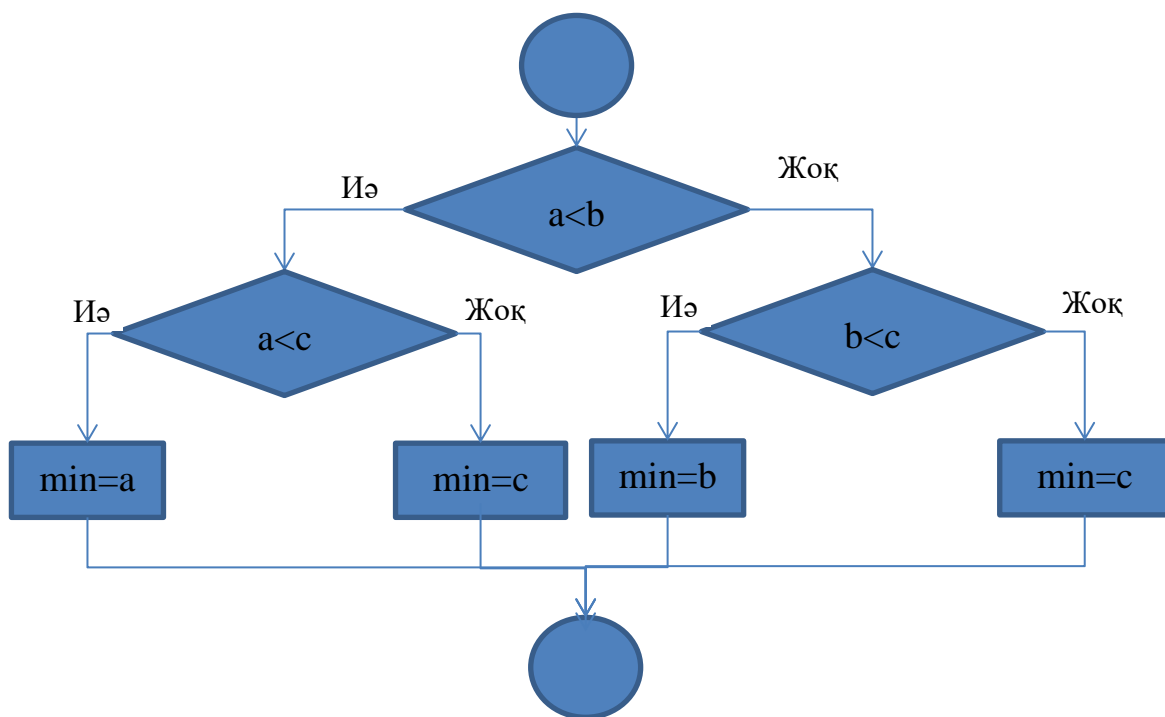
Сурет 35 – Екі санның арасындағы ең үлкен санды табу алгоритмы

Төменде листингты қараңыз:

```
a=int(input("Введите значение a="))
b=int(input("Введите значение b="))
if a>=b:
    max=a
else:
    max=b
print("Екі санның үлкені max=", max)
```

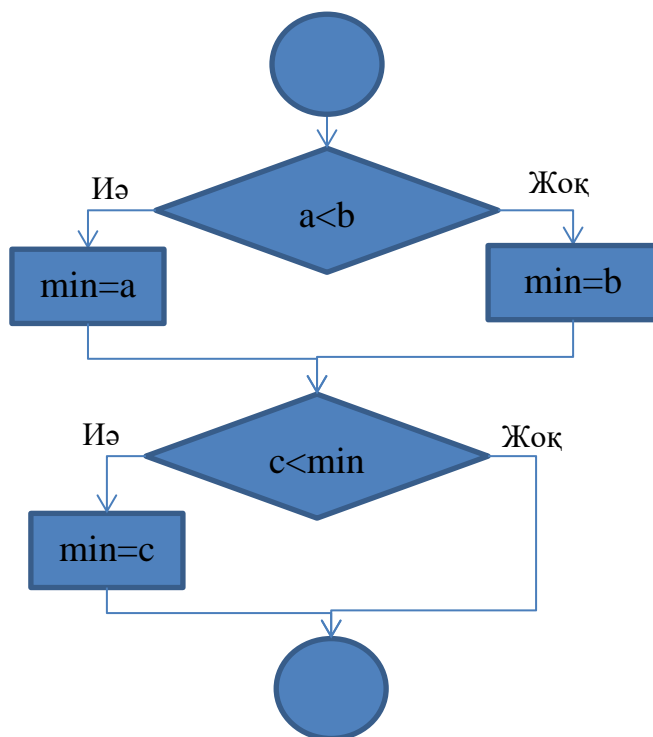
Есеп 4.2. Үш санның арасында ең кішісін табу .

Шешімі. Максимальды немесе минимальды элементті табу алгоритмін бірнеше жолмен бағдарламалауға болады. Есепті шешу алгоритмін (бірінші әдісі) құру фрагменті 36-ші суретте көрсетілген. Көптеген сандардың ішінен экстремалды мәнді таңдау керек болған жағдайда , алгоритм өте қиын және күрделі болуы мүмкін.



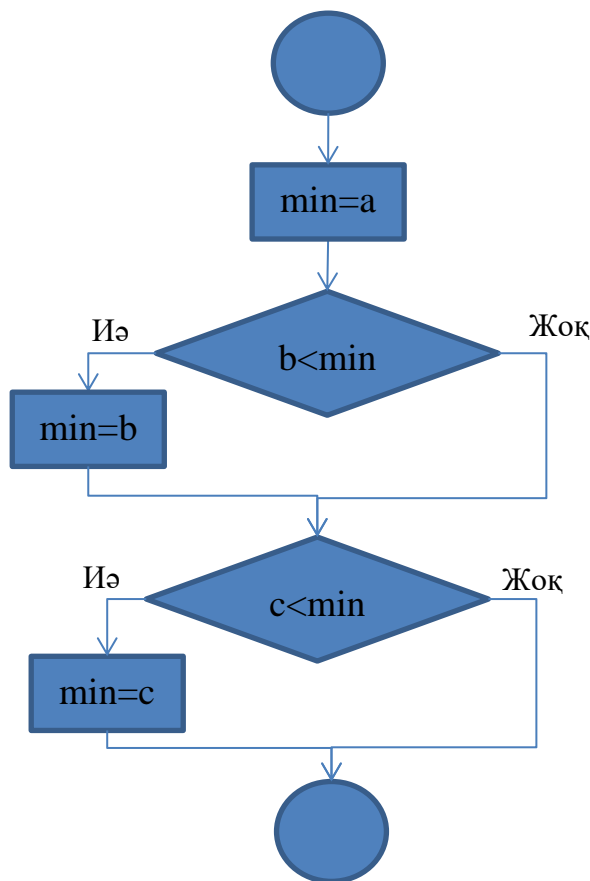
Сурет 36 – Үш санның арасынан ең кіші санды анықтау алгоритмы (бірінші әдіс)

37-ші суретте көрсетілген екінші әдіс фрагменті қарапайым және көрнекі. Екі санды бір-бірімен салыстырып, олардың ең азын анықтай отырып, әрбір келесі санды **min** ұяшығындағы санмен салыстырамыз және меншіктеу операторы ұяшықты қайта жазады.



Сурет 37 – Үш санның арасынан ең кіші санды анықтау алгоритмы (екінші әдіс)

Есепті шешу алгоритмін (үшінші әдіс) әзірлеу фрагменті 38-ші суретте көрсетілген.



Сурет 38 – Үш санның арасынан ең кіші санды анықтау алгоритмы (үшінші әдіс)

Үш санның арасынан ең кіші санды анықтау алгоритмы үшінші әдісі төмендегі листингте көрсетілген

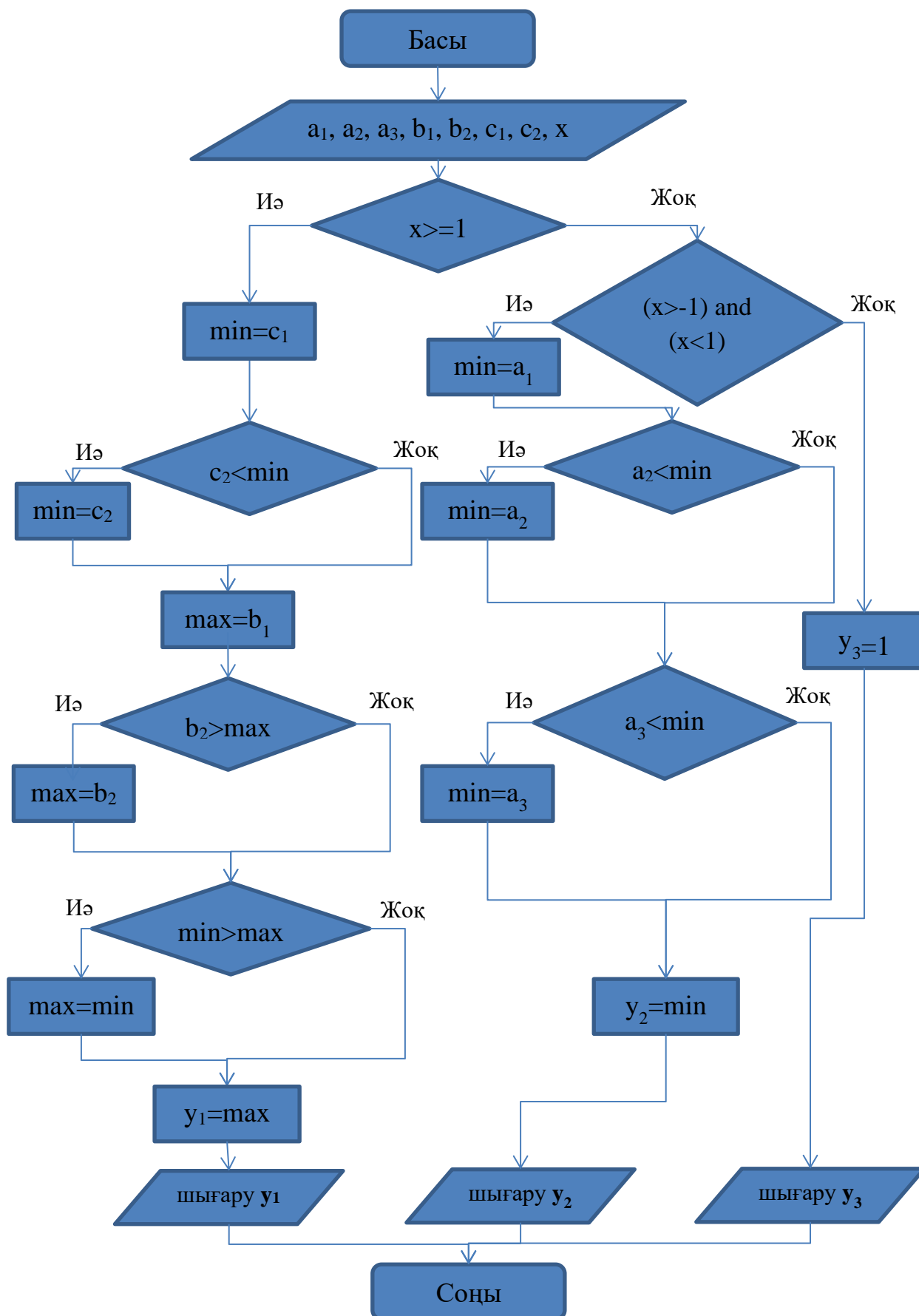
```
a=int(input("Введите значение a="))
b=int(input("Введите значение b="))
c=int(input("Введите значение c="))
min=a
if b<min:
    min=b
if c<min:
    min=c
print("Үш санның арасындағы min=", min)
```

Қарастырылған алгоритмдер негізінде біз келесі есептерді шешеміз.

Есеп 4.3. Егер келесі функция берілген болса, y функциясының мәнін есептеңіз.

$$y = \begin{cases} \min(a_1, a_2, a_3), & \text{если } -1 < x < 1 \\ \max(b_1, b_2, \min(c_1, c_2)), & \text{если } x \geq 1 \\ 1, & \text{если } x \leq -1 \end{cases}$$

Шешімі. Есепті шешу алгоритмін құру 39-ші суретте көрсетілген.



Сурет 39 – 4Алгоритм решения задачи 4.3

Төменде мәселені шешуге жауап беретін бағдарлама коды берілген. Листингте жасалған шегіністерге тағы бір рет назар аударған жөн. Әрбір **if** операторында есепті шешу алгоритміне сәйкес кірістірілген операторлардың өз тіркемесі және тиісінше оңға шегінісі болады. Шартты оператор өз жұмысын аяқтағаннан кейін келесі оператор(операторлар) шегініссіз жазылады(жазады), мысалы, келесі үзіндіні қараңыз:

```
if min>max:  
    max=min  
y1=max  
print("Тест 1 ветки")  
print("y1=", y1)
```

Max=min операторын орындағаннан кейін, **if** шартты операторы өз жұмысын аяқтайды, содан кейін операторлар (курсивпен көрсетілген) енді **if** операторына жатпайды және бағдарламаны терген кезде сол жаққа ауысады.

```
a1=int(input("Введите значение a1="))  
a2=int(input("Введите значение a2="))  
a3=int(input("Введите значение a3="))  
b1=int(input("Введите значение b1="))  
b2=int(input("Введите значение b2="))  
c1=int(input("Введите значение c1="))  
c2=int(input("Введите значение c2="))  
x=float(input("Введите значение x="))  
if x>=1:  
    min=c1  
    if c2<min:  
        min=c2  
    max=b1  
    if b2>max:  
        max=b2  
    if min>max:  
        max=min  
    y1=max  
    print("Тест 1 ветки")  
    print("y1=", y1)  
elif (x>-1) and (x<1):  
    min=a1  
    if a2<min:  
        min=a2  
    if a3<min:  
        min=a3  
    y2=min  
    print("Тест 2 ветки")
```

```
    print("y2=", y2)
else:
    y3=1
    print("Тест 3 ветки")
    print("y3=", y3)
```

4.6 Тармақталу алгоритмін қолдануға арналған тапсырмаларды түсіндіру

Есеп 4.1. Операторлар тобын орындағаннан кейін **s** айнымалысының мәні қандай болады?

```
n=2.5
f=0.5
d=True
s=0
if n<f:
    s=12
if f>=n:
    s=28
if d:
    s=39
print("s=", s)
```

Шешімі. Бұл жаттығуда **s** айнымалысының мәні 39-ға тең. Біз келесі түрде талқылаймыз. Алғашқы екі логикалық өрнек **жалған**, сондықтан үшінші логикалық өрнек тексеріледі, **d** ұяшығында **True**(ақиқат) мәні сақталады, сондықтан **s=39** операторы орындалады.

Есеп 4.2. Операторлар тобын орындағаннан кейін **j** айнымалысының мәні қандай болады?

```
w=3
p=5
j=3.5
if (j<p) and (j>w):
    j=j+0.5
    j=j+10
else:
    j=11
print("j=", j)
```

Шешімі. Біз келесі түрде талқылаймыз. **W**, **p**, **j** айнымалыларының мәндерін қойып және логикалық өрнекті тексеріп, біз оның мәні **True**(ақиқат) екенін аламыз. Сондықтан **j=j+0.5** операторы орындалады.;

содан кейін $j=j+10$ операторы орындалады. Жаттығуда келесі жауап: айнымалы мән $j=14.0$.

Есеп 4.3. Операторлар тобын орындағаннан кейін j айнымалысының мәні қандай болады?

```
j=3
k=15
m=20
if j<=k:
    if m>k:
        j=k%2
        j=j%3
    else:
        j=10
print("\n Значение j=", j)
```

Шешімі . K , m , j айнымалыларының мәнін енгізіп және екі логикалық өрнекті тексеру арқылы біз олардың мәні **True**(ақиқат) екенін білеміз. Демек, $j=k\%2$ операторы орындалады:, содан кейін: $j=j\%3$. Жаттығудағы жауап: j айнымалысының мәні **1** - ге тең.

Есеп 4.4. Операторлар тобын орындағаннан кейін j айнымалысының мәні қандай болады?

```
j=6
k=6
if j>k:
    j=j+2
    j=j+3
else:
    j=k-3
    j=j+4
print("\n Значение j=", j)
```

Шешімі. K және j айнымалыларының мәндерін қойып, логикалық өрнекті тексере отырып, оның мәні **False**(жалған) екенін білдік. Сондықтан , $j=k-3$ және $j=j+4$ операторлары орындалады. Жаттығудағы жауап: j айнымалысының мәні **7** - ге тең.

Есеп 4.5. Операторлар тобын орындағаннан кейін j айнымалысының мәні қандай болады?

```
w=3
p=5
```

```

j=3.5
if (j<p) and (j>w):
    j=j+0.5
    j=j+12
else:
    j=11
print("\n Значение j=", j)

```

Шешімі. Айнымалылардың мәндерін енгізіп отырып, біз екі қарапайым шарттың (**j<p**) және (**j>w**) ақиқат екеніне көз жеткізе аламыз, сондықтан барлық логикалық өрнек **True** болады. Осылайша: **j=j + 0.5** операторы содан кейін: **j=j+12** орындалады. Орындалғаннан кейін **j** ұяшығында **16.0** саны пайда болады.

Есеп 4.6. Шартты оператор орындалғаннан кейін **j** айнымалының мәні қандай болады

```

j=7
k=7
f=10
if j>=k:
    if f<=k:
        k=30%5
        j=(j%2)*k
    else:
        j=1
else:
    j=0
print("\n Значение j=", j)

```

Шешімі. Айнымалылардың мәндерін бірінші логикалық өрнекке қойып **True**(ақиқат) нәтижесін аламыз. Айнымалылардың мәндерін екінші логикалық өрнекке қойып біз **False** (жалған) нәтижесін аламыз, сондықтан оператор **else** тармағында орындалады (оператордың оң жағындағы шегініс бірінші деңгейдегі **if** операторы үшін **else** тармағына сәйкес келеді, сондықтан **j=0** орындалмайды, өйткені бірінші логикалық өрнекте **True** нәтижесі болғандықтан). Осылайша, жаттығуда қойылған сұраққа дұрыс жауап: **J** ұяшығында **бірлікке** тең мән болады.

Есеп 4.7. Операторлар тобын орындағаннан кейін **f** айнымалының мәні қандай болады ?

```

x=55
y=5e1
d=False

```

```

f=0
if d:
    f=x%2
if x<y:
    f=x
if x>y:
    f=int(2.9)
print("\n Значение f=", f)

```

Шешімі. Бірінші логикалық өрнекті тексеріп, нәтиже жалған екенін аламыз - **d** ұяшығында False мәні сақталады. Сондықтан **f=x%2** операторы орындалмайды. Екінші логикалық өрнектің ақиқат немесе жалған екенін анықтау үшін Сіз 5e1 саны өзгермелі нүкте түрінде жазылғанын және 50-ке тең екенін білуіңіз керек. Сондықтан екінші логикалық өрнек жалған және **f=x** операторы орындалмайды. Үшінші логикалық өрнекті тексеріп, біз оның дұрыс екеніне көз жеткіземіз, өйткені **55>50**. **F=int (2.9)** операторында бүтін түрге түрлендіру қолданылады. Осылайша, *шартты оператор* орындалғаннан кейін **f** ұяшығында **2** саны болады.

Есеп 4.8. Операторлар тобын орындағаннан кейін **j** айнымалының мәні қандай болады ?

```

j=10
k=10
if j>k:
    j=k-3
else:
    k=k-3
    j=k-3
print("\n Значение j=", j)

```

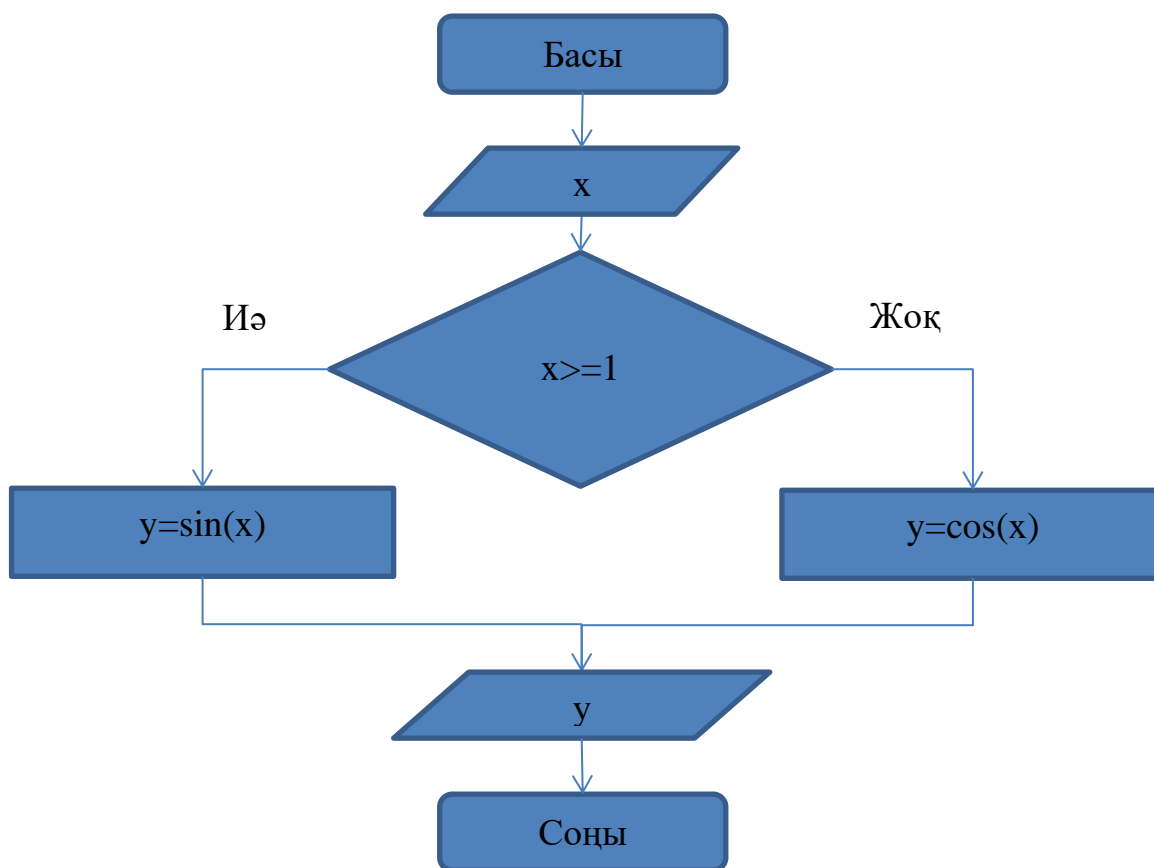
Шешімі. Логикалық өрнекті тексеріп, оның жалған екеніне көз жеткізіңіз. Демек, операторлар **else** тармағында орындалады. **K=k-3** операторы орындалғаннан кейін **k** ұяшығында **7** мәні болады, ал **j=k-3** операторы орындалғаннан кейін **j** ұяшығында **4** саны болады. Жаттығудағы жауап: **j** тең 4.

4.7 Есепті шығару мысалдары

Есеп 4.4. Берілген y функцияның мәнін табыңыз:

$$y = \begin{cases} \sin x, & \text{если } x \geq 1 \\ \cos x, & \text{если } x < 1 \end{cases}$$

Шешімі. 40-ші суретте есептің шешу алгоритмы блок-схемада көрсетілген.



Сурет 40 – 4.4 есептің шешу алгоритмы

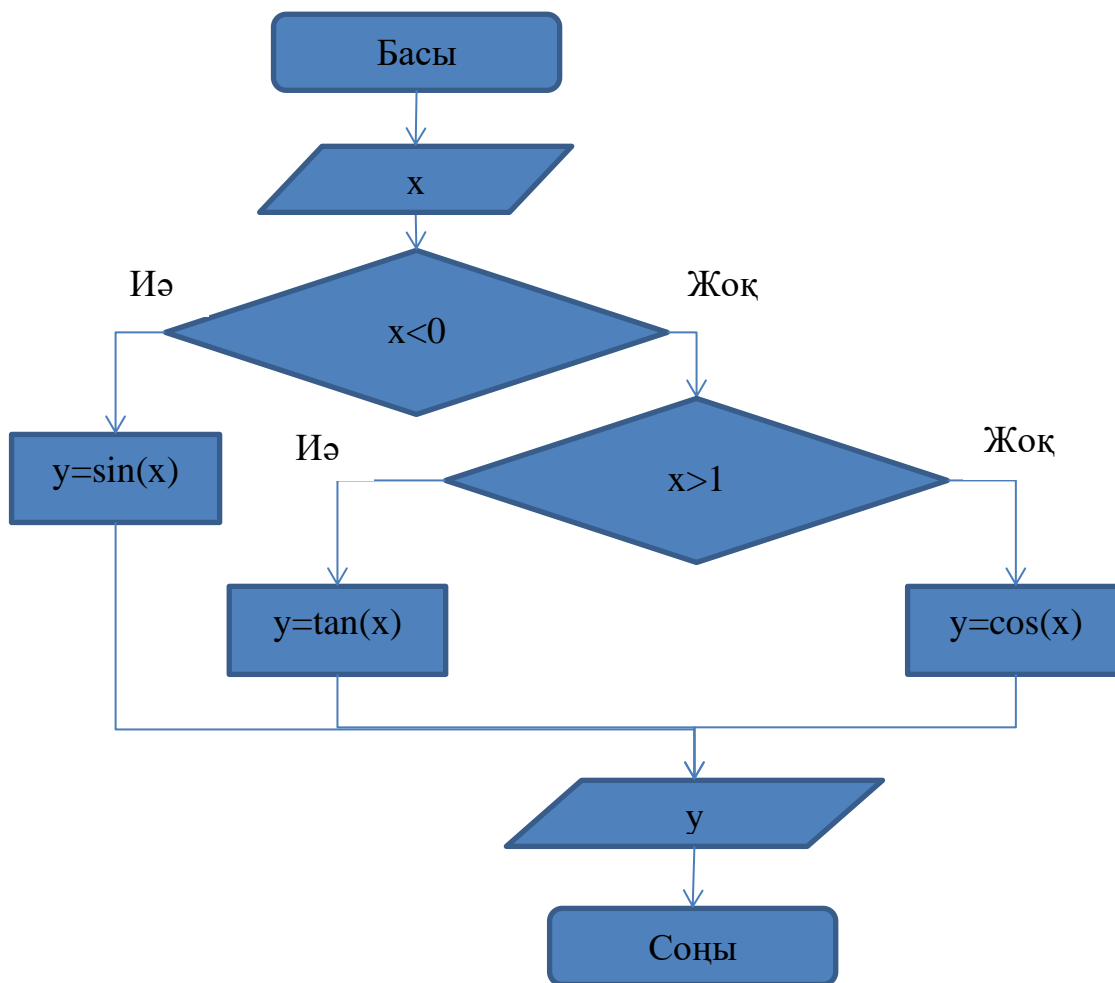
Листингте есепті шешуге жауап беретін бағдарлама коды орналасқан

```
from math import *
x=float(input("Введите значение x="))
if x>=1:
    y=sin(x)
else:
    y=cos(x)
print("\n Результат:", y)
```

Есеп 4.5. Берілген y функцияның мәнін табыңыз:

$$y = \begin{cases} \sin x, & \text{если } x < 0 \\ \cos x, & \text{если } 0 \leq x \leq 1. \\ \operatorname{tg} x, & \text{если } x > 1 \end{cases}$$

Шешімі. 41-ші суретте есептің шешу алгоритмы блок-схемасы көрсетілген



Сурет 41 – 4.5 есептің шешу алгоритмы

Листингте есепті шешуге жауап беретін бағдарлама коды орналасқан

```

from math import *
x=float(input("Введите значение x="))
if x<0:
    y=sin(x)
elif x>1:
    y=tan(x)
else:
    y=cos(x)
print("\n Результат:", y)
  
```

4.8 Бақылау сұрақтары

1. Таармақталған алгоритм деген не?
2. Қарапайым шартты оператор блок-схемаларында қалай жазылады?
3. Қарапайым шартты оператор бағдарламаларда қалай жазылады?
4. Қарапайым шартты оператор қалай жұмыс істейді?
5. Қысқартылған шартты оператор блок-схемаларында қалай жазылады?

6. Қысқартылған шартты оператор бағдарламаларда қалай жазылады?
7. Қысқартылған шартты оператор қалай жұмыс істейді?
8. Құрама шартты оператор блок-схемаларында қалай жазылады?
9. Құрама шартты оператор бағдарламаларда қалай жазылады?
10. Құрама шартты оператор қалай жұмыс істейді?
11. Блок-схемаларда көп мәнді тармақтар қалай жазылады?
12. Бағдарламаларда көп мәнді тармақтар қалай жазылады?
13. Шартты оператор бірнеше шарттарды тексеру кезінде қалай жұмыс істейді?

4.9 Өздігінен шешуге арналған есептер

1. Формула бойынша у мәнін есептеу алгоритмі мен бағдарламасын құрыңыз

$$Y = \begin{cases} X, & \text{если } X \leq 0 \\ 2X, & \text{если } X > 0 \end{cases}$$

2. Пайдаланушы пернетақтадан енгізген оң мәндерді ғана жинақтайтын(қосындысын) алгоритм мен бағдарламаны жасаңыз.

3. Үш айнымалының мәндері берілген. Нөлге тең мәндер санын есептейтін операторлар тізбегін жазыңыз. Осы есепті шешудің алгоритмі мен бағдарламасын құрыңыз.

4. Пайдаланушы енгізген санға байланысты "сізде қанағаттанарлық үлгерім бар", "сізде жақсы үлгерім бар", "сізде өте жақсы үлгерім бар" деген жауаптың үш түрін көрсететін алгоритм мен бағдарламаны құрыңыз.

5. Квадраттық теңдеудің түбірлерін формулалар ақылы табыңыз, қойылған есепке алгоритм және бағдарлама құрыңыз :

$$D = b^2 - 4ac,$$

$$x_1 = \frac{-b + \sqrt{D}}{2a}, \quad x_2 = \frac{-b - \sqrt{D}}{2a}.$$

6. Келесі есепті шешудің алгоритмі мен бағдарламасын құрыңыз. "Бару және қайту" теміржол билетінің құнын анықтау», егер баратын жерге дейінгі қашықтық және онда болу ұзақтығы белгілі болса, егер қашықтық 1000 км-ден асса, ал болу ұзақтығы 7 күннен асса, онда теміржол компаниясы 30% жеңілдік береді.

7. Келесі есептің шешудің алгоритмі мен бағдарламасын құрыңыз. Үш саны a,b,c берілген, олардың реттілігімен қатаң өсетінін(a<b<c), қатаң төмендейтінін(a>b>c) немесе осы екі шарттың бірде-біреуі орындалмайтынын тексеру қажет.

8. Келесі есептің шешу алгоритмі мен бағдарламасын құрыңыз. Үш санның арасында ең кішісін анықтау керек.

9. Келесі есептің шешу алгоритмі мен бағдарламасын құрыңыз. Функция у мәнін есептеңіз:

$$y = \begin{cases} 0, & \text{если } x < 0 \\ x, & \text{если } 0 \leq x \leq 1 \\ 1, & \text{если } x > 1 \end{cases}$$

10. Келесі есептіңі шешу алгоритмі мен бағдарламасын құрыңыз. Пайдаланушы екі санды енгізеді. Енгізілген сандардың кішісі 0 санымен, ал егер олар тең болса, 100 санымен ауыстырылады.