

6 ЦИКЛДЫҚ АЛГОРИТМ WHILE

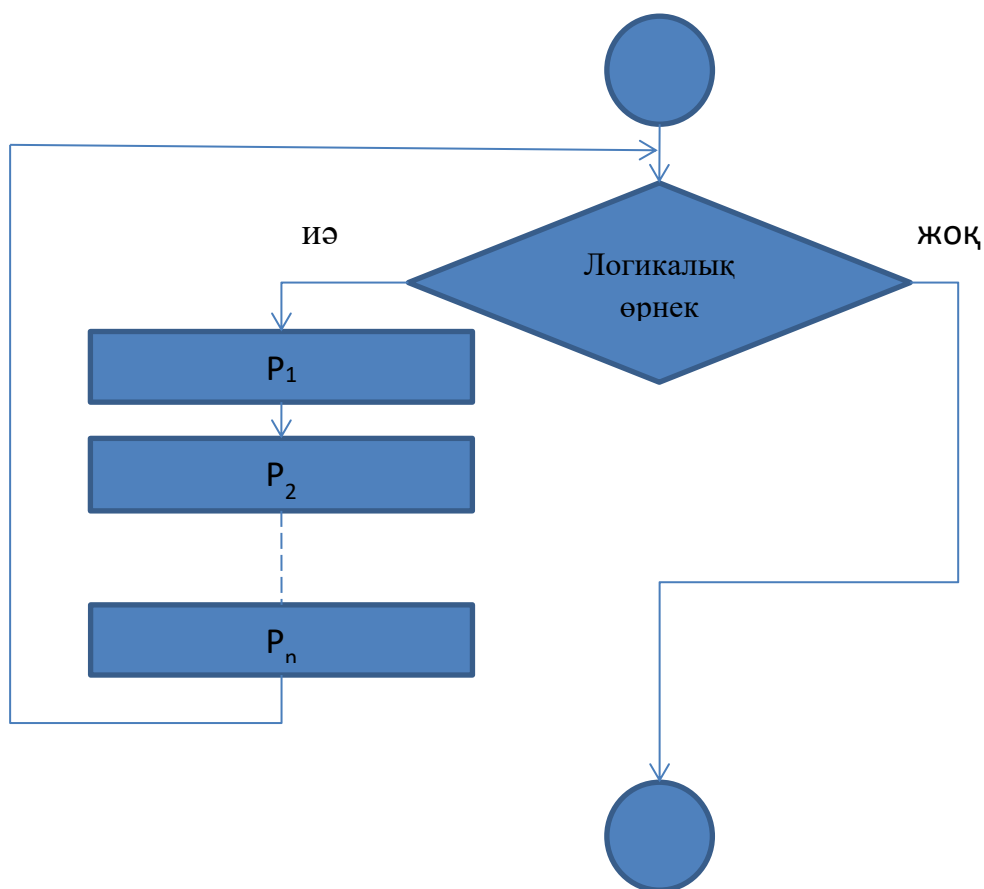
6.1 Цикл операторы while

For операторын орындау үшін цикл операторы (операторлары) қанша рет орындалатынын анықтайтын параметрлерді орнату қажет. **For** операторына баламалы болып қайталау саны белгісіз онда логикалық өрнек белгілі бір мәнді қабылдағанға дейін оператор(операторлар) орындалады.

Циклді қайталау саны алдын-ала белгісіз және алгоритмді орындау барысында ғана анықталатын циклдік құрылым **итеративті** деп аталады.

Мұндай циклдарды қанша рет цикл қайталанатынын нақты біле алмайтын тапсырмаларда қолданылуы керек. Мысалы, пайдаланушы қандай да бір бағдарламамен жұмыс істеуді бастау үшін парольді енгізуі керек. Ол қанша әрекетті қолданады? Белгісіз.

Сондықтан, мұндай міндеттерді жүзеге асыру үшін оларды шешудің тиісті әдістеріне ие болу керек. Python тілінде цикл құрылысын белгісіз қайталанулармен жүзеге асыру үшін **while** операторы немесе шартты алдын-ала тексеретін цикл қолданылады (57-ші сурет).



Сурет 57 – **while** операторының алгоритмының блок-схемасы

while операторының синтаксисі келесідей:

Бастапқы мәнді инициализациялау

while логикалық өрнек:

P_1

P_2

P_n

мұндағы P_1, P_2, \dots, P_n - операторлар; **while** (әзірге, соған дейін) – **Python** тілінің қызмет сөзі.

Егер **while** қызметтік сөзден кейінгі логикалық өрнек **True**(Ақиқат) болса, онда P_1, P_2, \dots, P_n операторлары орындалады, содан кейін логикалық өрнекті қайта тексеріледі. Егер логикалық өрнек **False**(жалған) болса, онда циклден шығады(цикл тоқтайды). Егер цикл тақырыбындағы жағдай басынан ақиқат болмаса, **while** циклі орындалмайды. Цикл құрылымындағы **for** операторының параметрі үнсіз келісім бойынша орнатылады, **while** циклінде жоқ болғандықтан, онда цикл параметрінің құру қажет болады. Сонымен қатар, бағдарламалау кезінде циклға дейін оның инициализациясын қамтамасыз ету қажет, ал циклде бұл айнымалы мәнді белгілі бір қадамға ұлғайту қажет.

Мысалы, төмендегі Листингте, "Студент"сөзі енгізілгенге дейін, **while** операторы бар цикл деректерді өңдейді.

```
name=""
while name !='Студент':
    name=input("Введите любое слово для печати или слово Студент для
выхода: ")
    if name!='Студент':
        print("Ответ = ", name)
```

Есеп 6.1. . 1-ден 50-ге дейінгі бүтін сандардың қосындысын ,**while** цикл операторын қолдана отырып табыңыз.

Шешімі. Бұрын осы есепті оператор **for**-ды пайдаланып шештік. Ең алдымен, бұл жағдайда кез-келген айнымалы циклде 1-ден 50-ге дейін өзгеруі керек. Цикл параметрі сияқты мән **while** құрылымында жоқ болғандықтан, бұл мысалда **k** айнымалысы осындай рөл атқарады. Осылайша, біз барлық елу шартты тұжырымдаймыз және жауапта 1275 санын аламыз. Есепті шешу алгоритмінің блок-схемасы 58-ші суретте көрсетілген.

Листингте есептің шешуге жауап беретін бағдарлама коды жазылған:

```
k=0
```

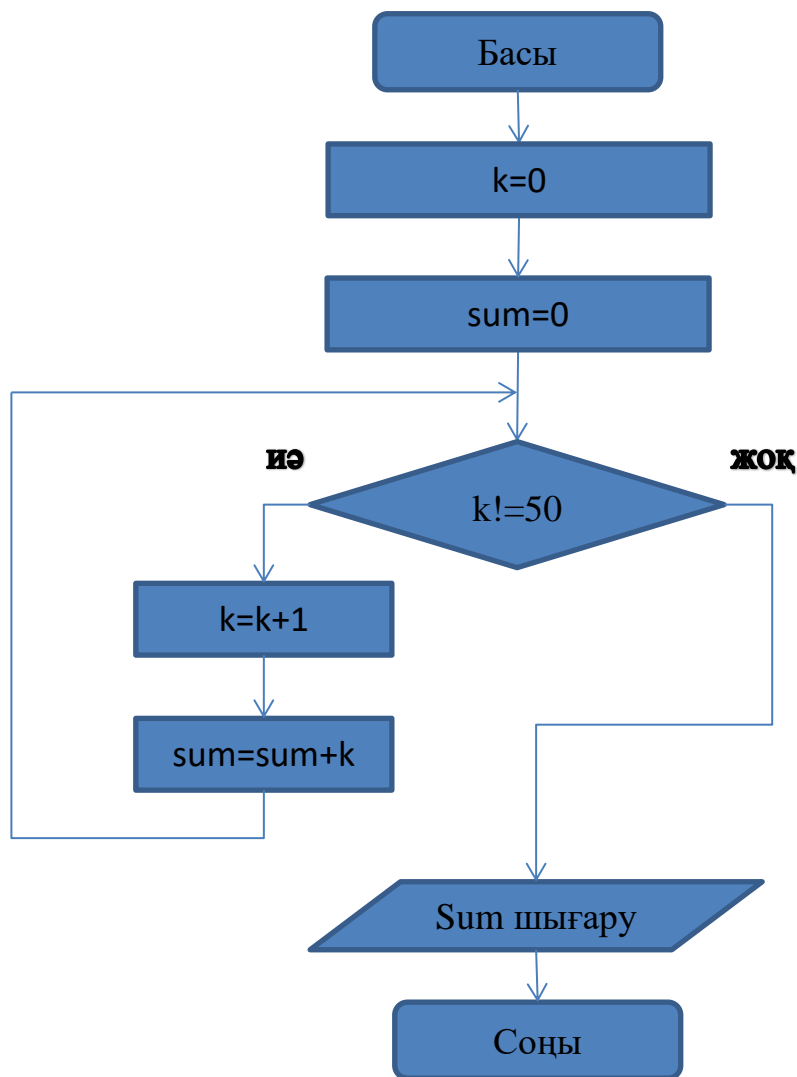
```
sum=0
```

```
while k!=50:
```

```
    k=k+1
```

```
    sum=sum+k
```

```
print("Сумма чисел от 1 до 50 sum = ", sum)
```



Сурет 58 – 6.1 есептің шешу алгоритмінің блок-схемасы

Функциялардың мәндерін немесе кез-келген сандық тізбекті (мысалы, арифметикалық прогрессия) есептеу кезінде олар көбінесе, арнайы жиынтықтар түрінде жазылады **қатарлар** деп аталатын. Көптеген сандарды, функцияларды, сандық әдістердің алгоритмдерін қатарлар немесе итерациялық алгоритмдер арқылы жазуға болады, бұл олардың жуық мәндерін берілген дәлдікпен есептеуге мүмкіндік береді. Мұндай тапсырмаларды бағдарламалау кезінде циклды ұйымдастыру үшін итерациялық әдістер қолданылады, онда кейбір **рекуррентті формула** есептеледі.

Рекурренттік формула (лат. recurrens, recurrens — қайта оралатын), келтіру формуласы — тізбектің бастапқы n мүшесі белгілі болған кезде оның кез келген мүшесін есептеуге мүмкіндік беретін $a_{n+p} = F(n, a_n, a_{n+1}, \dots, a_{n+p-1})$ түріндегі қатысты айтады, яғни қандай да болмасын тізбектің (көбінесе, сан тізбегінің) n -мүшесін алдыңғы мүшелері арқылы табатын формула.

Жалпы жағдайда бұл формула келесідей:

$$S_n = S_{n-1} + U_n,$$

мұнда S_n алғашқы n қатарының қосындысы, ол қосындының алғашқы қадамында есептеледі S_{n-1} ; U_n - ағымдағы қадамда алынған қосынды.

6.2 While циклдерін қолдануға арналған тапсырмалар мысалдарын түсіндіру

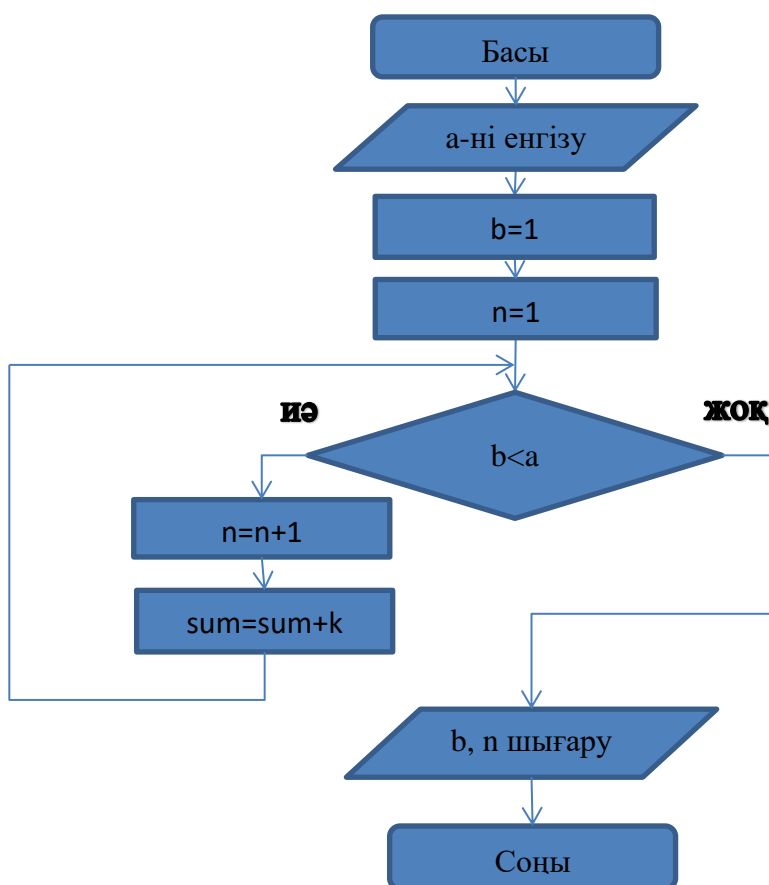
Итеративті циклдар мен рекуренттік(қайталанатын) қатынастарды қолданатын бірқатар тапсырмаларды қарастырамыз.

Тапсырма 6.1. Келесі сандар арасында $1, 1 + \frac{1}{2}, 1 + \frac{1}{2} + \frac{1}{3}, \dots$ енгізілген a айнымалысынан үлкен санды табыңыз.

Шешімі . Бұл тапсырманы шешу алгоритмі шексіз тізбектің мүшелерін есептеу алгоритмдеріне жатады. Біз шексіз тізбектің келесі мүшесін b деп белгілейміз. Оның нөмірі бөлшектің бөліміне сәйкес келсе, тізбектің келесі мүшесінің мәнін алу үшін алдыңғы мүшеге қосып n деп белгілеңіз. Сонда тізбектің келесі мүшесін есептеуге арналған итерациялық формула келесі түрде болады:

$$b_n = b_{n-1} + \frac{1}{n}.$$

Есептің шешу алгоритмінің блок-схемасы 59-шы суретте көрсетілген.



Сурет 59 – 6.1 тапсырманың шешу алгоритмінің блок-схемасы

Төменде есептің шешуіне арналған программа коды берілген:

```

a=float(input("Введите число a = "))
b=1
n=1
print("\n ", n, " ", b)
while b<a:
    n+=1
    b=b+1/n
    print(" ", n, " ", b)
print("Первое число, превышающее значение a:", '{0:4f}'.format(b),
"Количество итераций:", n)

```

60-ші суретте **a** мәні **3.95**-ке тең болған кездегі бағдарламаның нәтижесі 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/00000603.py
Введите число a = 3.95

1 1
2 1.5
3 1.8333333333333333
4 2.0833333333333333
5 2.2833333333333333
6 2.4499999999999997
7 2.5928571428571425
8 2.7178571428571425
9 2.8289682539682537
10 2.9289682539682538
11 3.0198773448773446
12 3.103210678210678
13 3.180133755133755
14 3.251562326562327
15 3.3182289932289937
16 3.3807289932289937
17 3.439552522640758
18 3.4951080781963135
19 3.547739657143682
20 3.597739657143682
21 3.6453587047627294
22 3.690813250217275
23 3.73429151108684
24 3.7759581777535067
25 3.8159581777535068
26 3.854419716215045
27 3.8914567532520823
28 3.927171038966368
29 3.9616537975870574
Первое число, превышающее значение a: 3.961654 Количество итераций: 29
>>> |
Ln: 37 Col: 4

```

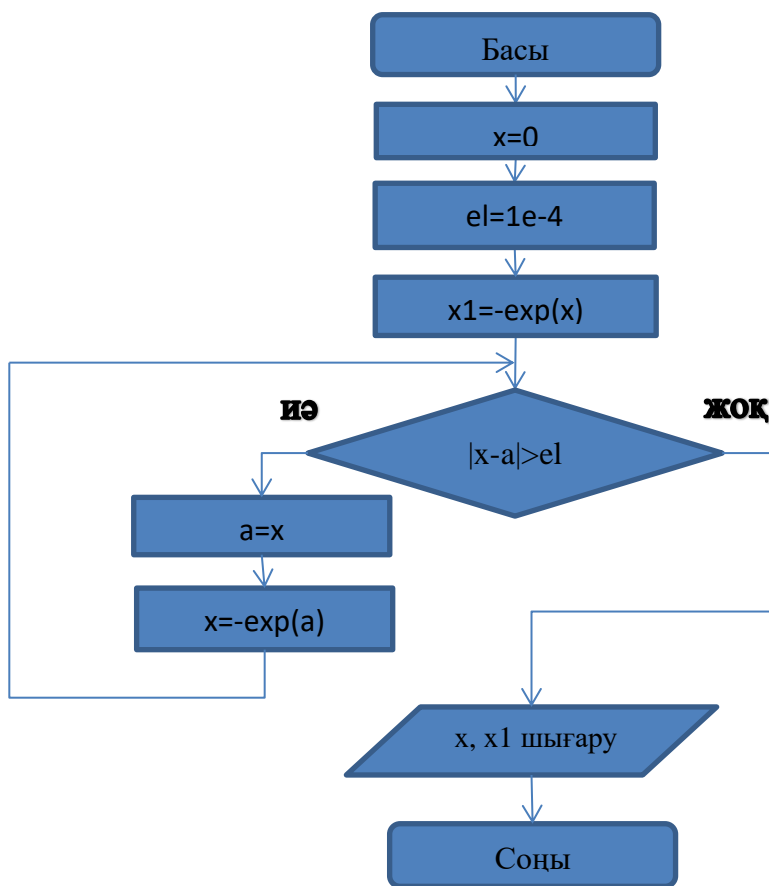
Сурет 60 – **a=3.95** тең болған жағдайдағы бағдарламаның нәтижесі

Тапсырма 6.2. Итерациялық формуланы $x_{i+1} = -e^{x_i}$ қолдана отырып,

$e^x + x = 0$ теңдеуінің түбірін $\varepsilon=10^{-4}$ дәлдікпен есептеңіз, мұндағы $i = 0, 1, 2, \dots; x_0 = 0$ Итеративті процесті келесі шарт орындалған $|x_{i+1} - x_i| < \varepsilon$ кезде аяқтаңыз.

Шешімі. Бұл тапсырманы шешу үшін x_{i+1} есептелген түбірдің келесі мәнінен x_i түбірінің алдыңғы мәнін алып тастау керек. Ол үшін циклдің әр қайталануында келесі x түбірді есептемес бұрын, ағымдағы x мәнін a айнымалысында сақтаймыз(сол кезде ол алдыңғы мән болады). Цикл x -тің айырмасы $\varepsilon = 10^{-4}$ кіші болған кезде аяқталады. Бағдарламада ε -ді $e1$ деп белгілейміз.

Тапсырманы шешу алгоритмінің блок-схемасы 61-ші суретте көрсетілген.



Сурет 61 – 6.2 тапсырманың шешу алгоритмінің блок-схемасы

Есептің бағдарламалық коды төменде көрсетілген:

```

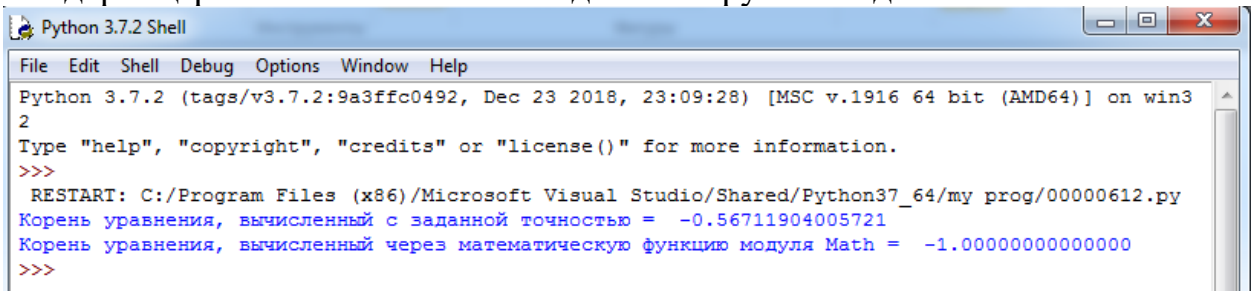
from math import *
a=1 #Инициализация значения a
x=0
x1=-exp(x)
e1=0.0001
while abs(x-a)>e1:
    a=x
  
```

```

x=-exp(a)
print("Корень уравнения, вычисленный с заданной точностью = ",
'{0:.14f}'.format(x))
print("Корень уравнения, вычисленный через математическую функцию
модуля Math = ", '{0:.14f}'.format(x1))

```

Бағдарлама жұмысының нәтижесі 62-ші суретте көрсетілген. Бұдан Math модулінің математикалық функциясы арқылы есептелген теңдеу түбірінің мәндерінің үлкен мәнге ие болатындығын көруге болады.



Сурет 62 –6.2 тапсырманың бағдарламма нәтижесі

Тапсырма 6.3. Берілген ε дәлдікпен e^x функциясының шамаланған мәнін есептейтін бағдарламаны жазыңыз.

Решение. e^x функциясын шексіз дәрежелі қатармен көрсетуге болады:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$$

EXP(x) функциясының мәнін ε дәлдігімен есептеу үшін, берілген қатардың барлық мүшелерінің қосындысынды модулі бойынша дәлдіктен асатын болу керек яғни ε -нан. Сондықтан қосындыны есептеу үшін циклдік процесті ұйымдастырып, шарт орындалғанша оны қайталау қажет

$$\left| \frac{x^n}{n!} \right| > \varepsilon.$$

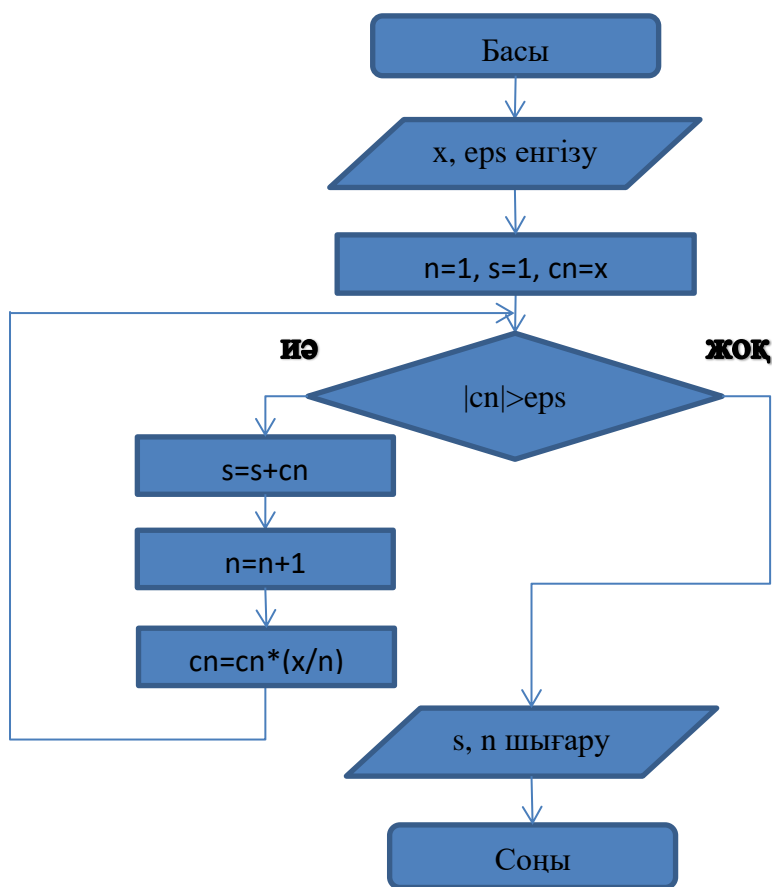
Жоғарыда келтірілген формуланы ескере отырып, қатардың көрші мүшелерін(оларды C_n және C_{n+1} деп белгілейміз) өзара қатынастың байланысты екенін көруге болады

$$C_{n+1} = C_n \cdot \frac{x}{n+1}, \quad \text{где } n = 0, 1, 2, \dots \text{ причём } C_0 = 1.$$

Осы қатынасты қолдана отырып, рет ретімен қатарларды есептеуге болады. Бұл жағдайда факториалды және дәрежені есептеу операциялары қажет емес. Қарастырылған дәрежелі қатардың қасиеті олардың жартылай қосындыларын есептеу үшін өте қарапайым және өте тиімді (есептеу жылдамдығын тиімділеу мағынасында) алгоритмдерді ұйымдастыруға мүмкіндік береді.

Тағы бір айта кететін жайт, алдыңғы мән негізінде кейінгі мәндерін есептеуге мүмкіндік беретін формулалар **рекурренттік**(қайталанатын)

қатынастар деп аталады. Тапсырманы шешу алгоритмінің блок-схемасы 63-ші суретте көрсетілген.



Сурет 63 – 6.3 тапсырманың шешу алгоритмінің блок-схемасы

Төменде есепті шешуге арналған бағдарламалық коды берілген

```
from math import *
x=float(input("Введите значение x = "))
eps=float(input("Введите точность вычислений eps = "))
s=1
n=1
cn=x
while abs(cn)>eps:
    s=s+cn
    n=n+1
    cn=cn*(x/n)
y=exp(x)
print("Значение функции, вычисленное с заданной точностью = ", s)
print("Значение функции, вычисленное через математическую функцию модуля Math = ", y)
```

64-ші суретте көрсетілген нәтижелерден функцияның қатардың қосындысы арқылы есептелген мәні жеткілікті дәлдікті беретінін көруге

болады. Ал ε мәні әлдеқайда аз болса, нәтиже $\text{EXP}(x)$ операторы арқылы есептелген мәндерге қарағанда дәлірек болады.

```

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 prog/00000613.py
Введите значение x = 0.7
Введите точность вычислений eps = 0.00000001
Значение функции, вычисленное с заданной точностью = 2.0137526991603205
Значение функции, вычисленное через математическую функцию модуля Math = 2.0137527074704766
>>>
Ln:9 Col:4

```

Сурет 64 – 6.3 тапсырманың бағдарламасының нәтижесі

Тапсырма 6.4. Кемімелі тізбекпен айнымалы таңбалы қатардың қосындысын берілген дәлдікпен ε есептеңіз.

$$\frac{(x-1)}{1!} - \frac{(x-1)^2}{2!} + \frac{(x-1)^3}{3!} - \dots + (-1)^n \frac{(x-1)^{n+1}}{(n+1)!} + \dots$$

Шешімі. Берілген ε дәлдікпен есептеу дегеніміз, қатардың келесі есептелген мүшесі ε санының абсолютті шамасынан аз болғанша, қатардың мүшелерін жинақтауды жалғастыру керек.

Көптеген тапсырмаларда келесі мүшені тікелей есептеу есептеу қиындықтарымен байланысты екенін ескеріңіз. Бұл жағдайда айнымалы мәнді ағымдағы қадамда $a_{n+1} = a_n \cdot q$ мәнін пайдаланып келесі қадамда есептеуге мүмкіндік беретін қайталанатын формуланы қолданған жөн - q үшін өрнекті a_{n+1}/a_n бөлу арқылы алуға болады.

Тапсырмада берілген қатарлар үшін қайталанатын формуланы шығарайық. Формуланың a_n мүшесі

$$a_n = (-1)^n \frac{(x-1)^{n+1}}{(n+1)!},$$

сонда a_{n+1} мүшесінің формуласы

$$a_{n+1} = (-1)^{n+1} \frac{(x-1)^{n+1+1}}{(n+1+1)!} = (-1)^{n+1} \frac{(x-1)^{n+2}}{(n+2)!}.$$

a_{n+1} мүшесін a_n - ге бөліп q үшін келесі өрнекті аламыз

$$q = \frac{a_{n+1}}{a_n} = \frac{(-1)^{n+1} \frac{(x-1)^{n+2}}{(n+2)!}}{(-1)^n \frac{(x-1)^{n+1}}{(n+1)!}} = \frac{(x-1)^{n+2} \cdot (-1)^{n+1} \cdot (n+1)!}{(-1)^n \cdot (x-1)^{n+1} \cdot (n+2)!} = -\frac{(x-1)}{n+2}.$$

Осылайша, берілген қатар үшін рекуренттік формуланың жалпы түрі

$$a_{n+1} = -a_n \frac{(x-1)}{n+2}.$$

Біздің жағдайымыз үшін n қатарының мүшесінің бастапқы мәні $n=0$ болады, өйткені бұл мәнді қатардың n мүшесінің формуласына ауыстырған кезде біз бірінші мүшенің мәнін аламыз $x-1$ немесе $a=x-1$ тең болатын.

$$a_n = (-1)^n \frac{(x-1)^{n+1}}{(n+1)!}$$

Программаның нәтижесі 65-ші суретте көрсетілген . Есептің шешімі төмендегі программа коды арқылы көрсетілген:

```

from math import *
x=float(input("Введите значение x = "))
e1=float(input("Введите точность вычислений e1 = "))
n=0
a=x-1
s=0
print("\n ", n, " ", a)
while abs(a)>e1:
    print("\n ", n, " ", a)
    s=s+a
    a=-a*(x-1)/(n+2)
    n=n+1
print("\n Значение итерации не учтенное в сумме ряда ", n, " ", a)
print("\n Сумма ряда, вычисленное с заданной точностью = ", s)

```

```

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/00000614.py
Введите значение x = 0.5
Введите точность вычислений e1 = 0.000001

0 -0.5
0 -0.5
1 -0.125
2 -0.020833333333333332
3 -0.0026041666666666665
4 -0.00026041666666666666
5 -2.170138888888889e-05
6 -1.5500992063492063e-06

Значение итерации не учтенное в сумме ряда 7 -9.68812003968254e-08
Сумма ряда, вычисленное с заданной точностью = -0.6487211681547619
>>>

```

Сурет 65 –6.4 тапсырманың программа нәтижесі

Тапсырма 6.5. Итерациялық формуланы пайдаланып $x_{i+1} = x_i - \frac{f(x)}{f'(x)}$ мұнда $i = 0, 1, 2, \dots$; $x_0 = 2,2$ теңдеу түбірін $f(x) = x^3 - 2x^2 + x - 3 = 0$ берілген дәлдік арқылы $\varepsilon = 10^{-5}$ есептеу керек. Итерациялық процесс келесі шарт орындалғанда аяқталады $|x_{i+1} - x_i| < \varepsilon$.

Шешімі. Тапсырманы шешіп, табылған түбірді теңдеуге қойып шешімнің дұрыстығын тексереміз.

Ол үшін $f'(x)$ туындысын есептейміз

$$f'(x) = F(x) = 3x^2 - 4x + 1.$$

Келесі белгілеулерді пайдаланамыз, **x** - түбірге ағымдағы жуықтау, **a** - алдыңғы жуықтау, **f** - алдыңғы мән үшін $f(x)$ функциясының мәні, **p** - алдыңғы мәні үшін $f'(x)$ туындысы, **i** - теңдеудің түбіріне ағымдағы жуықтау нөмірімен сәйкес келетін итерация нөмірі, **y**-берілген дәлдікпен табылған теңдеудің түбірі үшін $f(x)$ функциясының мәні.

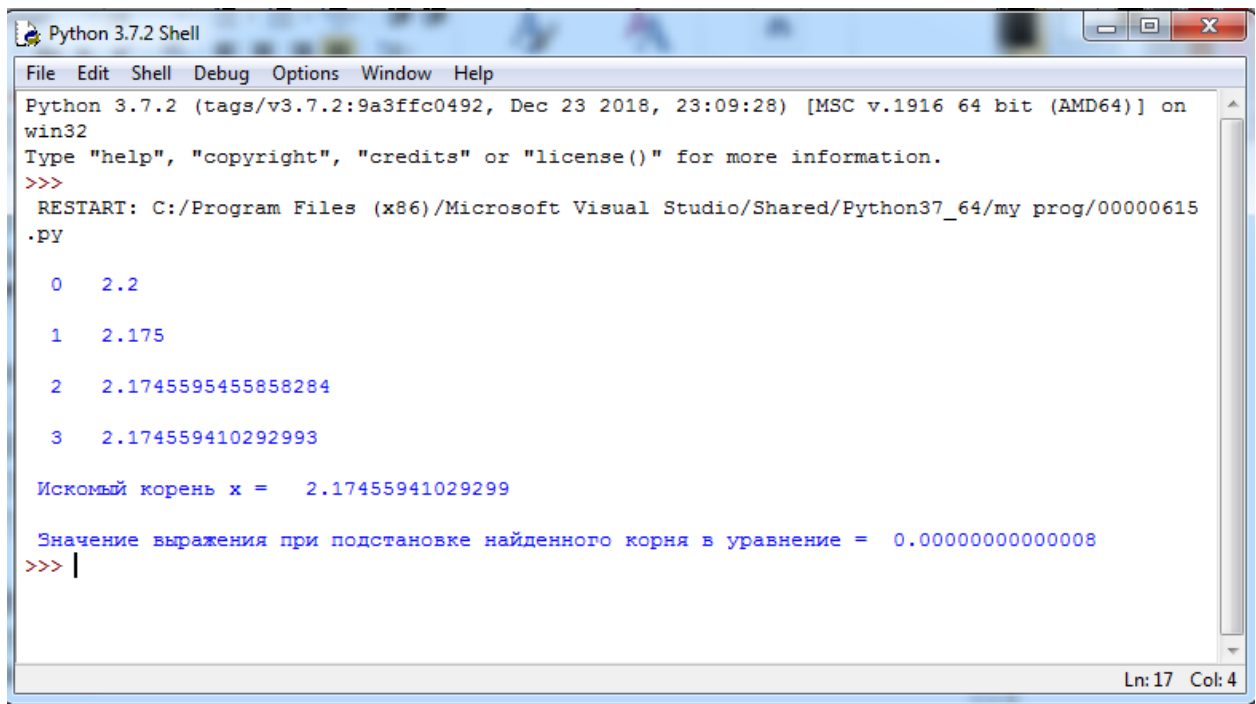
Егер ағымдағы және алдыңғы түбір мәндерінің арасындағы модуль айырмасы ε дәлдігінен аз болса, яғни біздің жағдайымыз үшін $|x-a| < \varepsilon$ орындалса, берілген дәлдік ε қамтамасыздандырылған, Тапсырманы шешуге жауап беретін бағдарлама коды төменде көрсетілген.

```

from math import *
x=2.2
e1=0.00001
i=0
a=1
print("\n ", i, " ", x)
while abs(x-a)>e1:
    a=x
    f=pow(x, 3)-2*a*a+a-3
    p=3*a*a-4*a+1
    x=a-f/p
    i=i+1
    print("\n ", i, " ", x)
y=pow(x, 3)-2*x*x+x-3
print("\n Искомый корень x = ", '{0:.14f}'.format(x))
print("\n Значение выражения при подстановке найденного корня в
уравнение = ", '{0:.14f}'.format(y))

```

Осылайша, 66-ші суретте келтірілген нәтижелерден **y** мәні табылған түбір теңдеуінде алмастыру ретінде табылғаны анық.



```
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 prog/00000615
.PY

0  2.2
1  2.175
2  2.1745595455858284
3  2.174559410292993

Искомый корень x =  2.17455941029299

Значение выражения при подстановке найденного корня в уравнение =  0.000000000000008
>>> |
```

Сурет 66 – 6.5 тапсырманың программасының нәтижесі

6.3 While Цикл жаттығуларындағы сұрақтар

Сұрақ 1. Операторлар тобын орындағаннан кейін у ұяшығында қандай мән болатынын анықтаңыз?

```
k=0
y=3
while k<y:
    y=y+1
    k=k+3
print("y = ", y)
```

Жауабы While операторында циклден шығу логикалық өрнек **False** (жалған) болған кезде болады. Бастапқы мәндерді қойып, біз бұл логикалық өрнек ақиқат екенін көреміз. Цикл операторларын орындау кезінде у ұяшығына төрт саны енгізіледі, ал k ұяшығына үш саны енгізіледі. Логикалық өрнек қайта тексеріледі ($3 < 4$). Оның нәтижесі: ақиқат. Цикл екінші өткен кезде у ұяшығына ($y=y+1$) **5** саны ал k ұяшығына ($k = k + 3$) **6** саны енгізіледі. Логикалық өрнекті тексеріп, оның **жалған** екеніне көз жеткіземіз ($6 < 5$), сондықтан циклден шығу орын алады. У ұяшығында **5** саны сақталады.

Сұрақ 2. Операторлар тобын орындағаннан кейін s ұяшығында қандай мән болатынын анықтаңыз?

```
s=7
i=1
```

```
while i*i==2:
    s=s+1/i
    i+=i
print("s = ", s)
```

Жауабы. Бұл мысалда мұқият болу керек және логикалық өрнек әрқашан **False**(жалған) мәнге ие болатындығын ескеру керек, сондықтан цикл денесі орындалмайды. **S** ұяшығының мәні өзгеріссіз қалады және **7** - ге тең болады.

Сұрақ 3. Операторлар тобын орындағаннан кейін **z** ұяшығында қандай мән болатынын анықтаңыз?

```
a=1
z=4
while a<=3:
    a+=a
    z=a+1
print("z = ", z)
```

Жауабы. Логикалық өрнек жалған болып, циклден шығу үшін **a** ұяшығында төрт немесе одан да көп сан болуы керек. Бұл жағдайда цикл бірінші өткен кезде **a** ұяшығында **2** мәні болады, ал **z** ұяшығында **3** саны болады. Осылайша, цикл тағы екі рет орындалады. **z=a+1** операторы **z** ұяшығының мәнін екіге көбейтеді. Соңғы жауап: **z = 5**.

Сұрақ 4. Операторлар тобын орындағаннан кейін **a** ұяшығында қандай мән болатынын анықтаңыз?

```
a=2
d=1
while (a+d)<=7:
    a=a+1
    d=d+1
a=a*d
print("a = ", a)
```

Жауап . Циклге алғаш кірген кезде **(a+d)** ұяшықта үш мәні болады. **a=a+1** операторы орындалғаннан кейін **a** мәні **3-ке** тең болады, ал **d** ұяшығында-**2-ге** тең мән болады(**d=d+1** операторы орындалғаннан кейін). Логикалық өрнекте тексерілетін осы екі ұяшықтың мәндерінің қосындысы **беске** тең болады. **5 < 7** шарт ақиқат, сондықтан цикл қайтадан орындалады. Бірлікке өсе отырып **a** және **d** ұяшықтарының мәндері сәйкесінше **төрт** және **үш** мәндеріне тең болады. Олардың қосындысы **жетіге** тең болады, яғни цикл үшінші рет орындалады. Бірлікке өссе отырып **a** ұяшығының мәні **бес**, ал **d**

ұяшығының мәні **төрт** болады. Логикалық өрнек: $5+4 \leq 7$ жалған болып шығады, циклден шығады және **$a=a*d$** операторы орындалады, ол жауабы болып табылады.

Сұрақ 5. Операторлар тобын орындағаннан кейін **n** ұяшығында қандай мән болатынын анықтаңыз?

```
n=2
x=1
while x<=4:
    x=x+1
    n=n+x
print("n = ", n)
```

Жауабы . Логикалық өрнек **False**(жалған) болған кезде **while** операторы циклден шығады. Бұл мысалда циклден шығу **x** ұяшығында төрт саннан үлкен сан болған кезде пайда болады. Циклде **x** ұяшығы оның мәнін бірлікке өсіре отырып байқалады, яғни циклден шығу **x** ұяшығында бес саны болған кезде пайда болады. Осылайша, циклде **n** ұяшығының мәнін **x** айнымалы мәнімен қосу арқылы есептеу керек, циклден шыққан кезде **n** ұяшығында **16** саны болады.

Сұрақ 6. Операторлар тобы орындалғаннан кейін **a** айнымалының мәнін анықтаңыз?

```
a=1
z=1
while a<=3:
    a=a+1
a=a+z
a=a+10
print("a = ", a)
```

Жауабы . Циклде **while** операторымен бір операторы орындалады **$a=a+1$** . Мұны циклде бір оператордың орындалуын білдіретін шегініс арқылы көруге болады. Циклден шығу болады егер **a** ұяшығы төртке тең болғанда(өйткені a циклдің әр қадамы үшін бірлікке артады). Циклден шыққаннан кейін **$a=a+z$** операторын орындағаннан кейін, **a** ұяшығында бес саны болады. Әрі қарай **$a=a+10$** операторы орындалады. Ұяшықтың **a**-ның мәні 15-ке тең болады.

Сұрақ 7. Операторлар тобы орындалғаннан кейін **s** ұяшығында қандай мән болатынын анықтаңыз?

```
a=20
d=5
while (a-d)>=10:
```

```
a=a-1
d=d+1
s=a+d
print("s = ", s)
```

Жауабы. Логикалық өрнекті тексеріп, оның ақиқат екеніне көз жеткіземіз. Әрі қарай, **a** ұяшығының мәні бірлікке азаяды, **d** ұяшығының мәні бірлікке артады. Осы процедуранан кейін әр уақытта ұяшық мәндерінің айырмасы $(a-d) \geq 10$ тексеріледі. **A** ұяшығында **17** саны, ал **d** ұяшығында **8** саны болған кезде циклден шығуға болады. **S=a+d** операторын орындағаннан кейін ұяшық мәні шығады **s = 25**.

Сұрақ 8. Операторлар тобы орындалғаннан кейін **a** айнымалының мәні неге тең болатынын анықтаңыз?

```
a=1
z=1
while z<=4:
    z+=z
    z=a+1
    a=z
print("a = ", a)
```

Жауабы. Бұл мысалда цикл денесінде үш оператор орындалады. Келесі заңдылықты орнатуға болады: **z** және **a** ұяшықтарының мәндері операторлар орындалған кезде тең болады, өйткені **z** ұяшығындағы мән **a** ұяшығына орналасып операторлар орындалғаннан кейін **a=z** болады. Циклден шығуға болады егер **z** ұяшығы бес санына тең болса, сондықтан **a** ұяшығында да бес саны сақталады, бұл осы жаттығудың жауап болып табылады.

Сұрақ 9. Операторлар тобы орындалғаннан кейін **p** ұяшығында қандай мән болатынын анықтаңыз ?

```
p=1
i=2
while p<9:
    p=p*i
    i=i+1
    p=p*i
print("p = ", p)
```

Жауабы. Циклде **p=p*i** операторы орындалады. Қалған операторлар цикл аяқталғаннан кейін орындалады. **i** ұяшықтың мәні циклден шыққанға дейін өзгеріссіз қалады және екіге тең болады, тек **p=p*i** операторы орындалғаннан кейін **p** ұяшығының мәні өзгереді. **P** ұяшығына **2, 4, 8, 16**

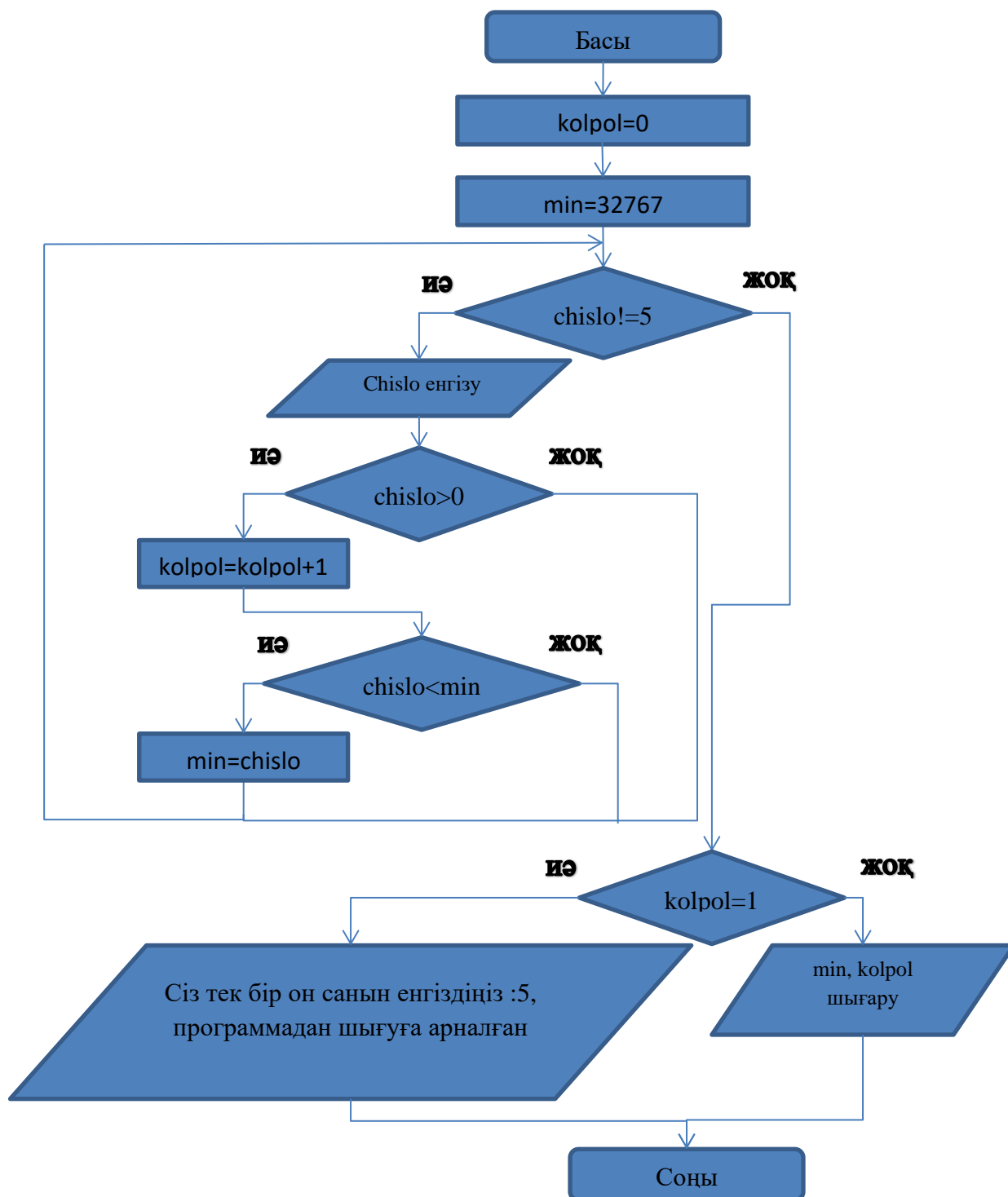
мәндері рет-ретімен енгізіледі. Бұл сандар p ұяшығында **16** саны болған кезде циклден шығу болатындығы көрсетеді. $I=i+1$ операторын орындағаннан кейін, i үшке тең болады. $P=p*i$ операторы орындалғаннан кейін, p ұяшығында **48** саны сақталады.

6.4 Есептерді шығару мысалдары

Есеп 6.4.1. Нақты сандар тізбегі енгізіледі. Тізбектің соңғы элементі беске тең екендігі белгілі. Оң сандардың санын және олардың ең кішісін табыңыз.

Шешімі. Есепті шешу алгоритмінің блок-схемасы 67-ші суретте көрсетілген. Логикалық өрнек жалған мәнді қабылдаған кезде **while** операторымен циклден шығу 67-ші суретте көрсетілген алгоритмге сәйкес жүзеге асырылады. Сондықтан, пайдаланушы 5 санына тең емес мәндерді енгізгенге дейін циклде оң сандар мен олардың ең кішісін табу үшін қажетті тиісті операторлар орындалады. Листингте есепті шешуге жауап беретін бағдарлама коды көрсетілген:

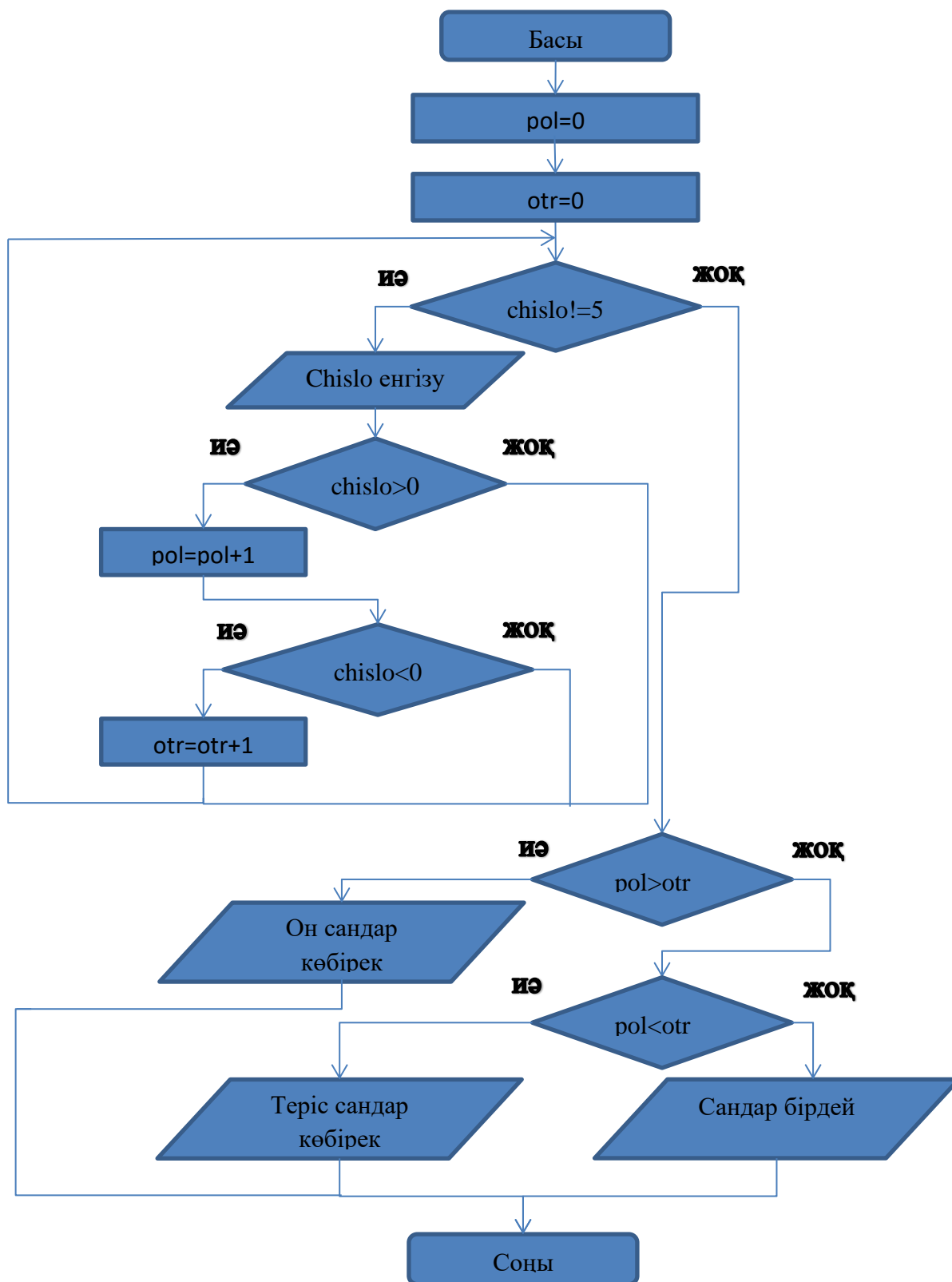
```
kolpol=0
min=32767
chislo=0
while chislo!=5: # Пока число, введенное в цикле, не равно числу 5,
выполняются операторы цикла
    chislo=float(input("Введите очередное число = "))
    if chislo>0: # Проверка на положительность очередного введенного
числа
        kolpol=kolpol+1
        if chislo<min: # Реализация алгоритма поиска минимального
положительного элемента
            min=chislo
    if kolpol==1:
        print("\n Вы ввели только одно положительное число: 5,
предназначенное для выхода из программы")
    else:
        print("\n Количество положительных чисел Kolpol = ", kolpol)
        print("\n Минимальное из них min = ", min)
```

Сурет 67 – 6.4.1 есептің блок-схемасының алгоритмы

Есеп 6.4.2. Нақты сандар тізбегі енгізіледі. Тізбектің соңғы элементі беске тең екендігі белгілі. Олардың қайсысы теріс немесе оң екенін анықтаңыз.

Шешімі. Есепті шешу алгоритмінің блок-схемасы 68-ші суретте көрсетілген. Шешім алгоритмі 6.4.1 есепті шешу алгоритміне өте ұқсас.



Сурет 68 – 6.4.2 есептің блок-схемасының алгоритмы

Листингте есепті шешуге жауап беретін бағдарлама коды көрсетілген:

```

pol=0
otr=0
chislo=0
while chislo!=5: # Пока число, введенное в цикле, не равно числу 5,
выполняются операторы цикла
    chislo=float(input("Введите очередное число = "))
    if chislo>0: # Проверка на положительность очередного введенного
числа
        pol=pol+1
    if chislo<0: # Проверка на отрицательность очередного введенного
числа
        otr=otr+1
    if pol>otr:
        print("\n Положительных чисел больше")
    else:
        if pol<otr:
            print("\n Отрицательных чисел больше")
        else:
            print("\n Чисел равное количество")
print("Положительных чисел = ", pol)
print("Отрицательных чисел = ", otr)

```

Есеп 6.4.3. Нөлге тең емес нақты сандар тізбегі енгізіледі. Тізбектің соңғы элементі беске тең екендігі белгілі. Соңғы енгізілген бес санын алып тастап, барлық теріс сандар мен оң сандардың қосындысын бөлек анықтаңыз.

Шешімі. Есепті шешу алгоритмінің блок-схемасы 69-ші суретте көрсетілген. Шешу алгоритмі 6.4.1 және 6.4.2 есептерді шешу алгоритмдеріне өте ұқсас.

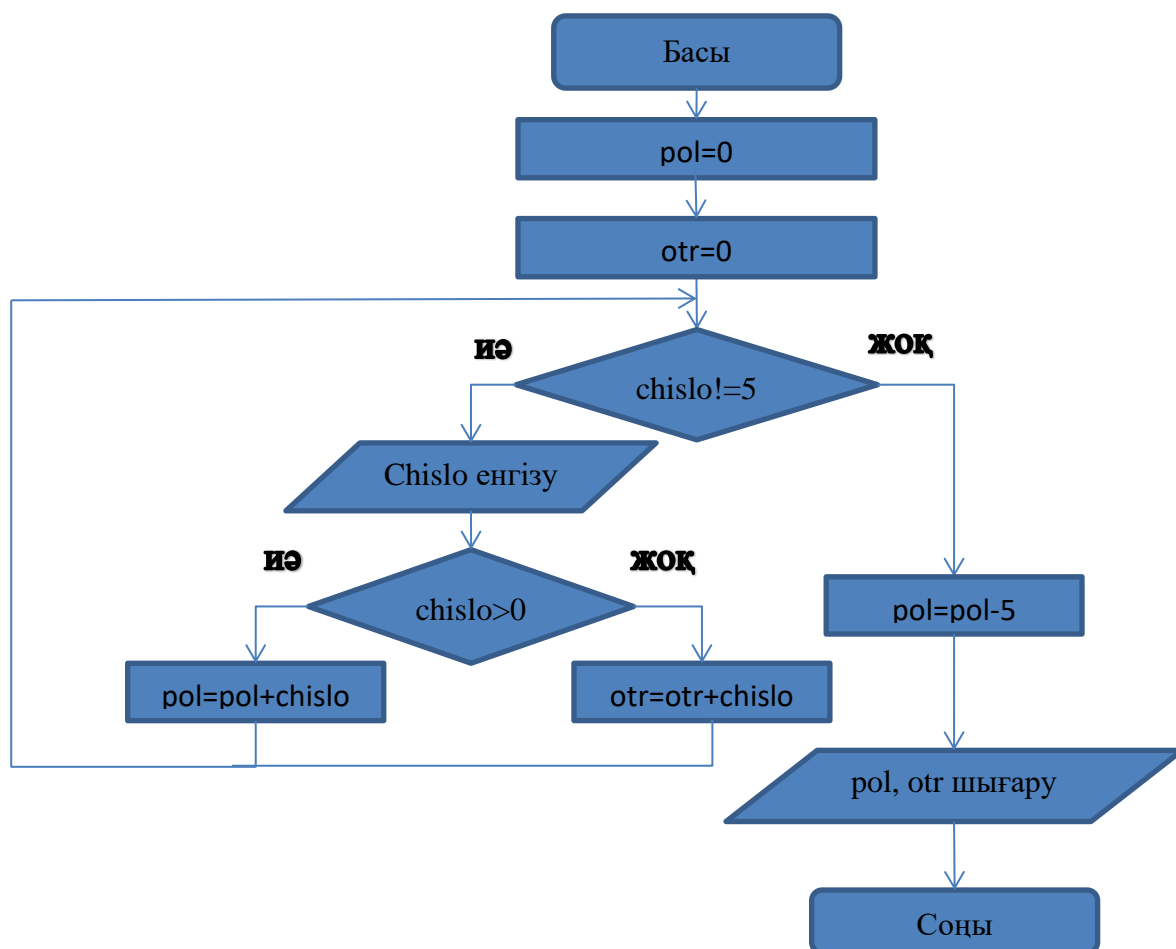
Листингте есепті шешуге жауап беретін бағдарлама коды көрсетілген:

```

pol=0
otr=0
chislo=0
while chislo!=5: # Пока число, введенное в цикле, не равно числу 5,
выполняются операторы цикла
    chislo=float(input("Введите очередное число = "))
    if chislo>0: # Проверка на положительность очередного введенного
числа
        pol=pol+chislo # Суммирование введенных положительных чисел
    else:
        otr=otr+chislo # Суммирование введенных отрицательных чисел
    pol=pol-5 # Согласно условию вычитаем из суммы последнее число
равное 5

```

```
print("\n Сумма положительных чисел равна = ", pol)
print("\n Сумма отрицательных чисел равна = ", otr)
```



Сурет 69 – 6.4.3 есептің блок-схемасының алгоритмы

Есеп 6.4.4. Нөлге тең емес бүтін сандар тізбегі енгізіледі. Тізбекті аяқтау үшін 0 енгізу керек екені белгілі. Осы сандардың арифметикалық ортасын табыңыз.

Шешімі. Есепті шешу алгоритмінің блок-схемасы 70-ші суретте көрсетілген. Соңғы сан шығу үшін **нөлге** тең болу керек және нөлге тең емес бүтін сандар тізбегіне жатпайтынын ескеру қажет. Жалпы, шешім алгоритмі 6.4.1 - 6.4.3 есептерін шешу алгоритмдеріне өте ұқсас.

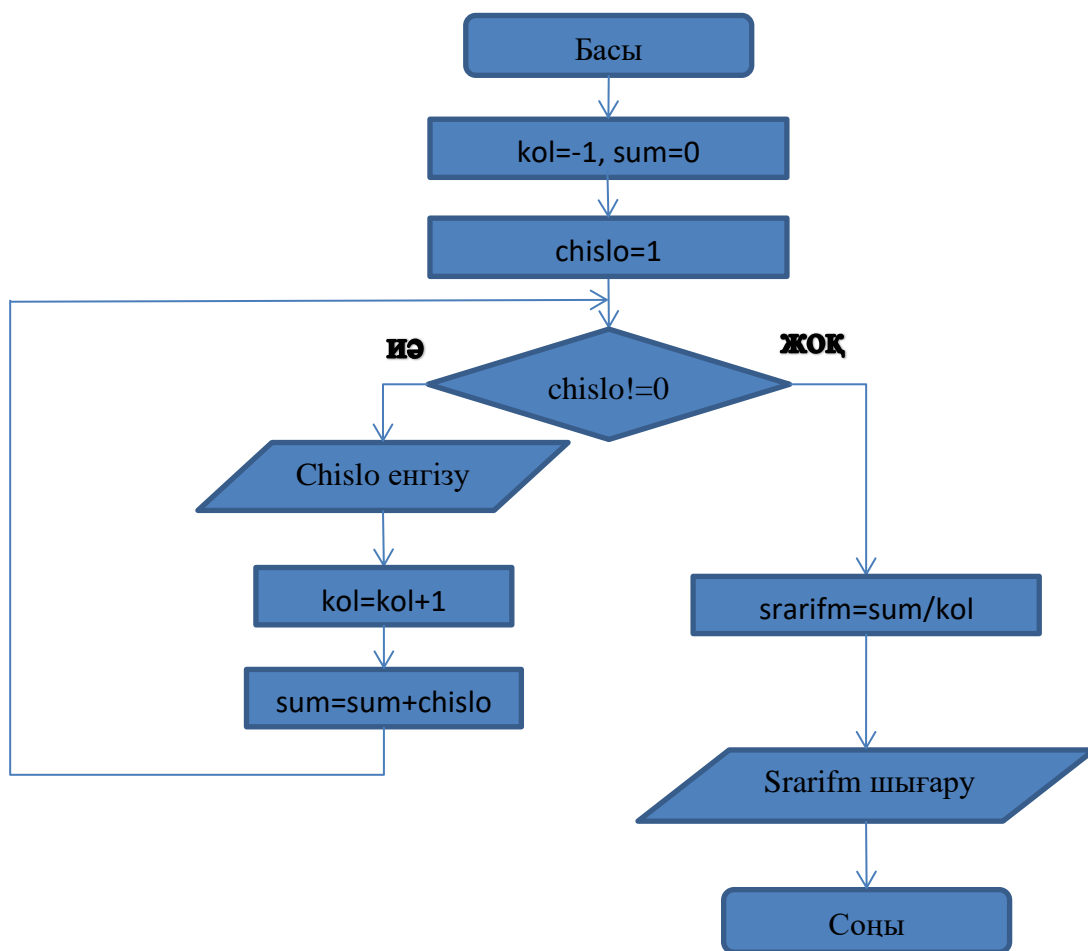
Листингте есепті шешуге жауап беретін бағдарлама коды көрсетілген:

```
kol=-1
sum=0
chislo=1
while chislo!=0: # Пока число, введенное в цикле, не равно числу 0,
    выполняются операторы цикла
    chislo=int(input("Введите очередное число "))
```

```

kol=kol+1
sum=sum+chislo
if kol==0:
    print("\n В последовательности нет ни одного числа")
    input("Нажмите ENTER, чтобы завершить программу")
else:
    srarifm=sum/kol
    print("\n Среднее арифметическое чисел равно = ", srarifm)
    print("\n Количество чисел = ", kol, "Сумма чисел чисел = ", sum)
    input("Нажмите ENTER, чтобы завершить программу")

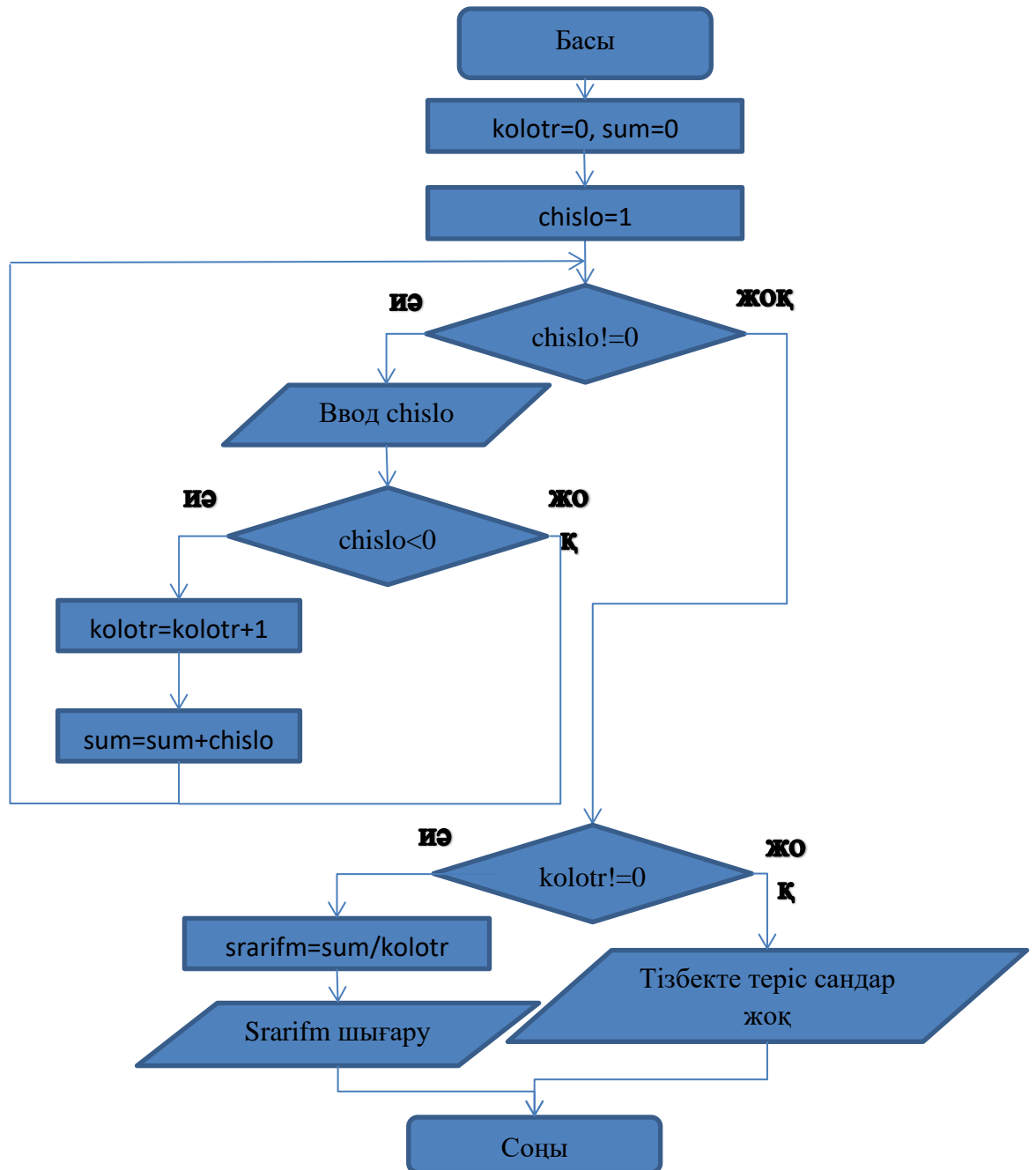
```



Сурет 70 – 6.4.4 есептің блок-схемасының алгоритмы

Есеп 6.4.5. Нөлге тең емес нақты сандар тізбегі енгізіледі. Тізбекті аяқтау үшін 0 санын енгізу керегі белгілі. Теріс сандардың санын және олардың арифметикалық ортасын табыңыз.

Шешімі. Есепті шешу алгоритмінің блок-схемасы 71-ші суретте көрсетілген. Шешу алгоритмі 6.4.1 - 6.4.4 есептерінің шешімдеріне ұқсас.



Сурет 71 – 6.4.5 есептің блок-схемасының алгоритмы

Листингте есепті шешуге жауап беретін бағдарлама коды көрсетілген:

```

kolotr=0
sum=0
chislo=1
while chislo!=0: # Пока число, введенное в цикле, не равно числу 0,
    выполняются операторы цикла
    chislo=float(input("Введите очередное число "))
    if chislo<0:
        kolotr=kolotr+1
        sum=sum+chislo
  
```

```

if kolotr!=0:
    srarifm=sum/kolotr
    print("\n Среднее арифметическое отрицательных чисел равно = ",
srarifm)
    print("\n Количество отрицательных чисел = ", kolotr, "Сумма
отрицательных чисел чисел = ", sum)
else:
    print("\n В последовательности нет отрицательных чисел")
input("Нажмите ENTER, чтобы завершить программу")

```

6.5 Бақылау сұрақтары

1. Белгісіз қайталанулар саны бар циклдар қандай жағдайларда қолданылатынын айтыңыз.
2. Қандай циклдік құрылымды итеративті деп санауға болады?
3. While цикл операторының алгоритмінің жалпы түрін көрсетіңіз.
4. While цикл операторының синтаксисін жазыңыз.
5. While цикл операторының жұмысы туралы айтып беріңіз. Мысалдар келтіріңіз.
6. Қайталанатын қатынастар қандай жағдайларда қолданылады? Қайталанатын формуланы шығару алгоритмі туралы айтып беріңіз.

6.6 Өздігінен шешуге арналған есептер

1. Келесі есептің шешу алгоритмі мен бағдарламасын құрыңыз. Максималды мәнін табыңыз $y = \cos x / \sin x$ функцияның $[a, b]$ кесіндісінде h қадамымен.
2. Нақты сандар тізбегі енгізіледі. Тізбектің соңғы элементі 5-ке тең. Оң сандардың санын және олардың ең кішісін табыңыз. Алгоритм мен бағдарламаны құрыңыз.
3. Бүтін сандар тізбегі енгізіледі. Тізбектің соңғы элементі 1 екендігі белгілі. Олардың қайсысы көбірек екенін анықтаңыз: оң немесе теріс. Алгоритм мен бағдарламаны құрыңыз.
4. Нөлге тең емес нақты сандар тізбегі енгізіледі. Тізбекті аяқтау үшін 0-ді енгізу керек. Бағдарламада барлық оң және барлық теріс сандардың қосындылары есептелуі керек. Алгоритм мен бағдарламаны құрыңыз.
5. Бүтін сандар тізбегі енгізіледі. Тізбекті аяқтау үшін 0-ді енгізу керек. Бірінші теріс санға дейінгі оң сандардың қосындысын анықтаңыз. Алгоритм мен бағдарламаны құрыңыз.
6. Бүтін сандар тізбегі енгізіледі. Тізбекті аяқтау үшін сізге 10-ды енгізу керек. Нөлге тең емес сандардың қосындысын есептеп, жауап ретінде осы қосындыны шығарыңыз. Алгоритм мен бағдарламаны құрыңыз.
7. Бүтін сандар тізбегі енгізіледі. Тізбекті аяқтау үшін сізге 100 санын енгізу керек. Бағдарламада қосындысы берілген саннан аз сандар санын анықтаңыз. Алгоритм мен бағдарламаны құрыңыз.

8. R клиентінің банкке алғашқы салымы және Pr жылдық кірісінің пайызы белгілі. Салым 1 миллион рубльден асатын мерзімді және осы салымның мөлшерін анықтаңыз. Алгоритм мен бағдарламаны құрыңыз.

9. 10-дан 30-ға дейінгі кесіндіде жұп сандардың қосындысын есептеңіз. Алгоритм мен бағдарламаны құрыңыз.

10. Ересектер үшін мінсіз салмақты формула бойынша анықтаңыз: мінсіз салмақ = бой-100. Циклден шығу: бойының мәні = 250. Алгоритм мен бағдарламаны құрыңыз.