

### 6.3 While Цикл жаттығуларындағы сұрақтар

**Сұрақ 1.** Операторлар тобын орындағаннан кейін  $y$  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  $y$  ұяшығына төрт саны енгізіледі, ал  $k$  ұяшығына үш саны енгізіледі. Логикалық өрнек қайта тексеріледі ( $3 < 4$ ). Оның нәтижесі: ақиқат. Цикл екінші өткен кезде  $y$  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.  $y$  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)>=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.

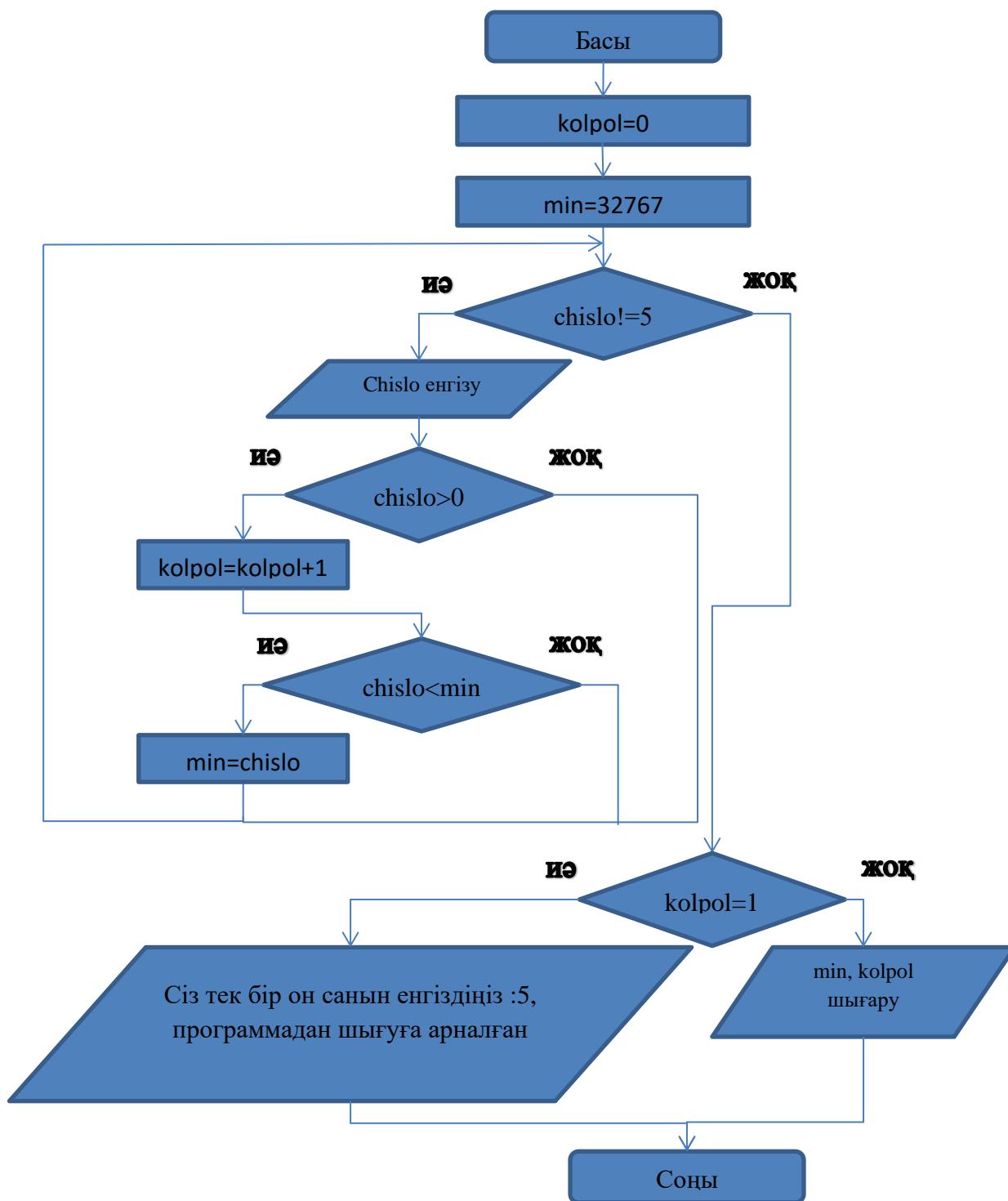
## 6.4 Есептерді шығару мысалдары

**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
```

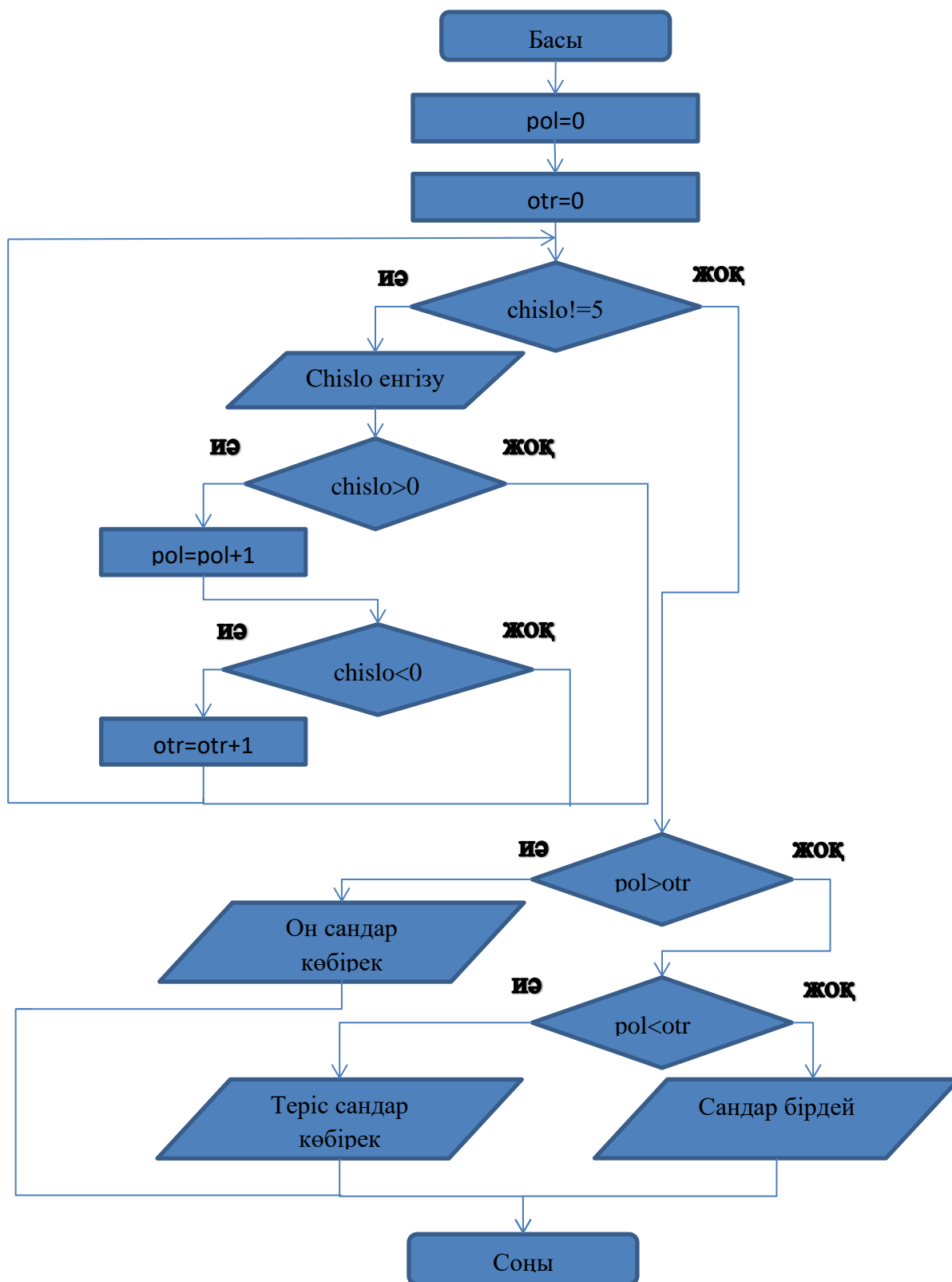
```
while chislo!=5: # Пока число, введенное в цикле, не равно числу 5,
выполняются операторы цикла
    chislo=float(input("Введите очередное число = "))
    if chislo>0: # Проверка на положительность очередного введенного числа
        kolpol=kolpol+1
        if chislo<min: # Реализация алгоритма поиска 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.

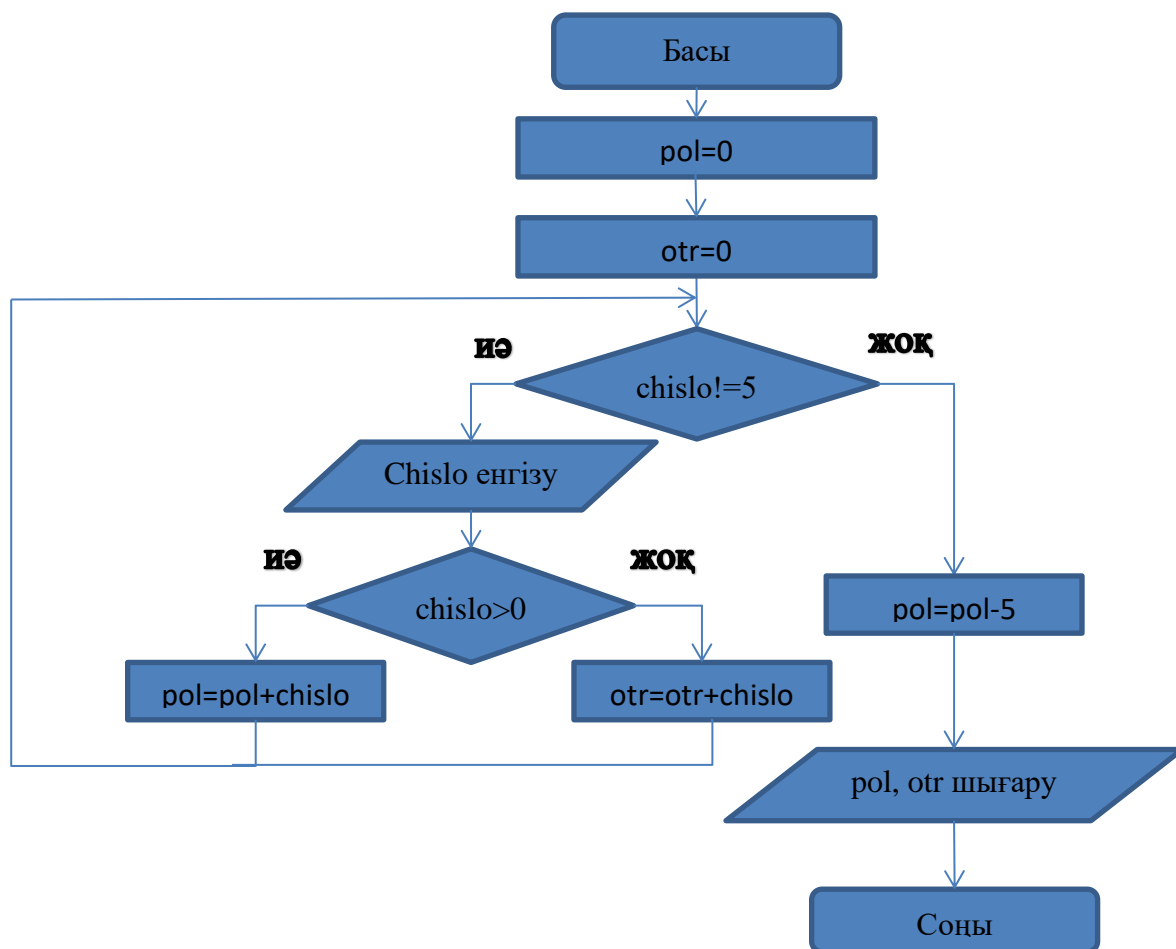
**Шешімі.** Есепті шешу алгоритмінің блок-схемасы 69-ші суретте көрсетілген. Шешу алгоритмі 6.4.1 және 6.4.2 есептерді шешу алгоритмдеріне өте ұқсас. Листингте есепті шешуге жауап беретін бағдарлама коды көрсетілген:

```

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:

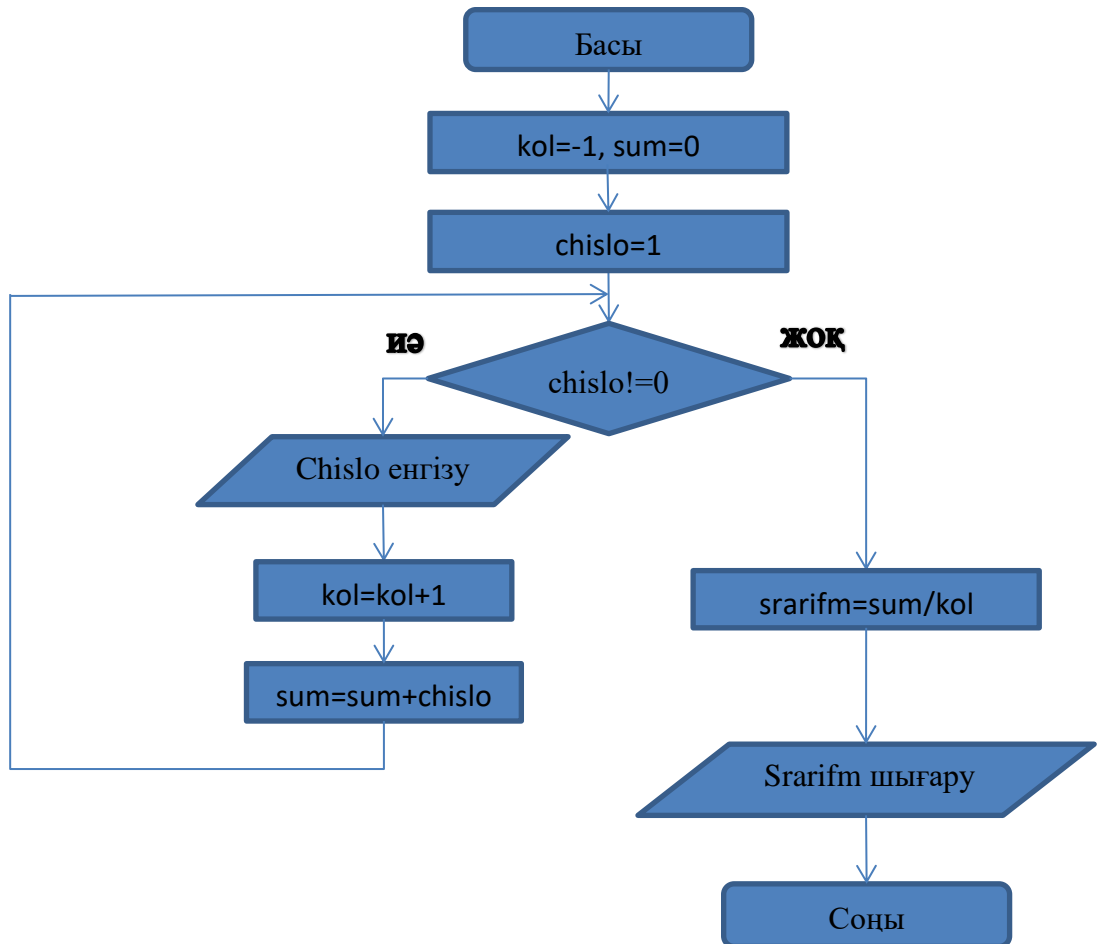
```

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")

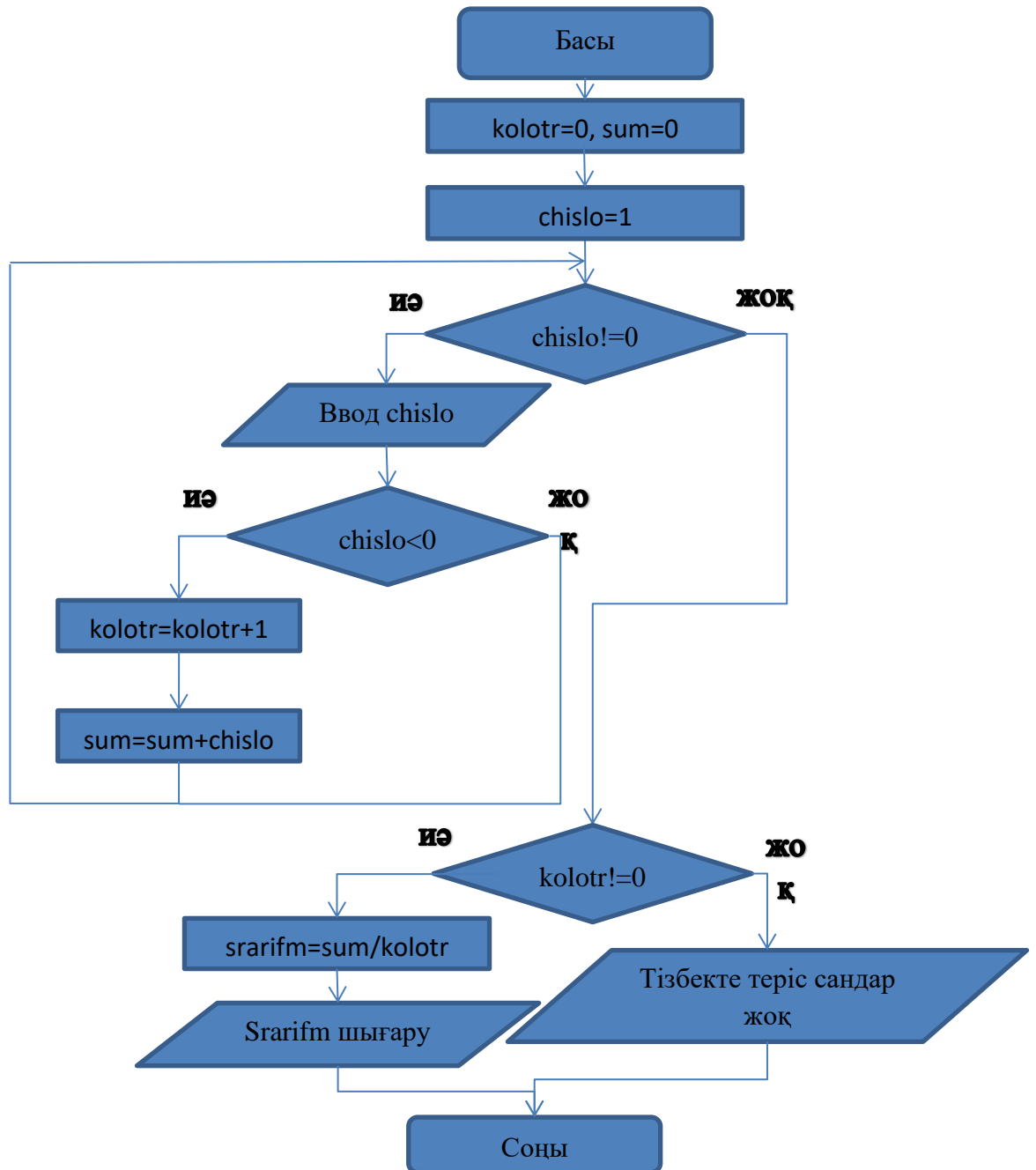
```



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 керекті 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 P<sub>r</sub> 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.