

7 КОРТЕЖДЕРМЕН ЖӘНЕ ТІЗІМДЕРМЕН ЖҰМЫС ІСТЕУ

7.1 Кортеждерді жариялау(хабарлау)

Python тіліндегі **кортеждер** массивтер деп аталатын құрылымдарға ұқсас. Массивтермен жұмыс істеу әдістері C++, Microsoft Visual Basic, Delphi және т.б. сияқты танымал бағдарламалау тілдерін оқытуда қарастырылады. Айта кету керек, оларда массивтер біртұтас болып саналатын бірнеше ұқсас объектілердің (сандар, таңбалар, жолдар және т. б.) ресми бірлестігі ретінде анықталады.

Python-да кортеж осы бағдарламалау тілінде жұмыс істеуге болатын тізбектің бір түрі ретінде анықталады. Басқа бағдарламалау тілдеріндегі массив мәліметтерінен түбегейлі айырмашылығы-бұл біріктірілген мәліметтер тізбегі **олардың мәндерін өзгертуге мүмкіндік бермейді**. Бірақ бұл жерде белгілі бір артықшылықтары да бар. Біріншіден, кортеж элементтерін өңдеу жылдамдығы артады, өйткені жүйе мәндер өзгермейтінін алдын-ала біледі. Екіншіден, кортеж сияқты құрылымды басқа құрылымдарды қалыптастыру үшін қолдануға болады: мысалы, сөздіктерде. Үшіншіден, мысалы, тізімдерге қарағанда аз жад алады, сонымен қатар, кортеж элементтері кездейсоқ өзгерістерден қорғалған.

Егер массив бір типтегі объектілердің жиынтығы болса, онда кортеж Python-да толық әртекті объектілер болуы мүмкін мысалы, жолдық және сандық мәндер немесе дыбыстық файлдар, бейнелеу файлдары сияқты.

Кортеж жариялау синтаксисі келесі түрде:

Кортеж аты = (элемент 1, элемент 2, ...элемент N)

Мысалы

```
korteg=( 1, 2, 3,4, 5)
```

Басқа жазба болуы мүмкін, мысалы, жол түріндегі мәндер үшін

Кортеж аты =(«Элемент1»,
«Элемент 2»,
«Элемент 3»,
«Элемент N»)

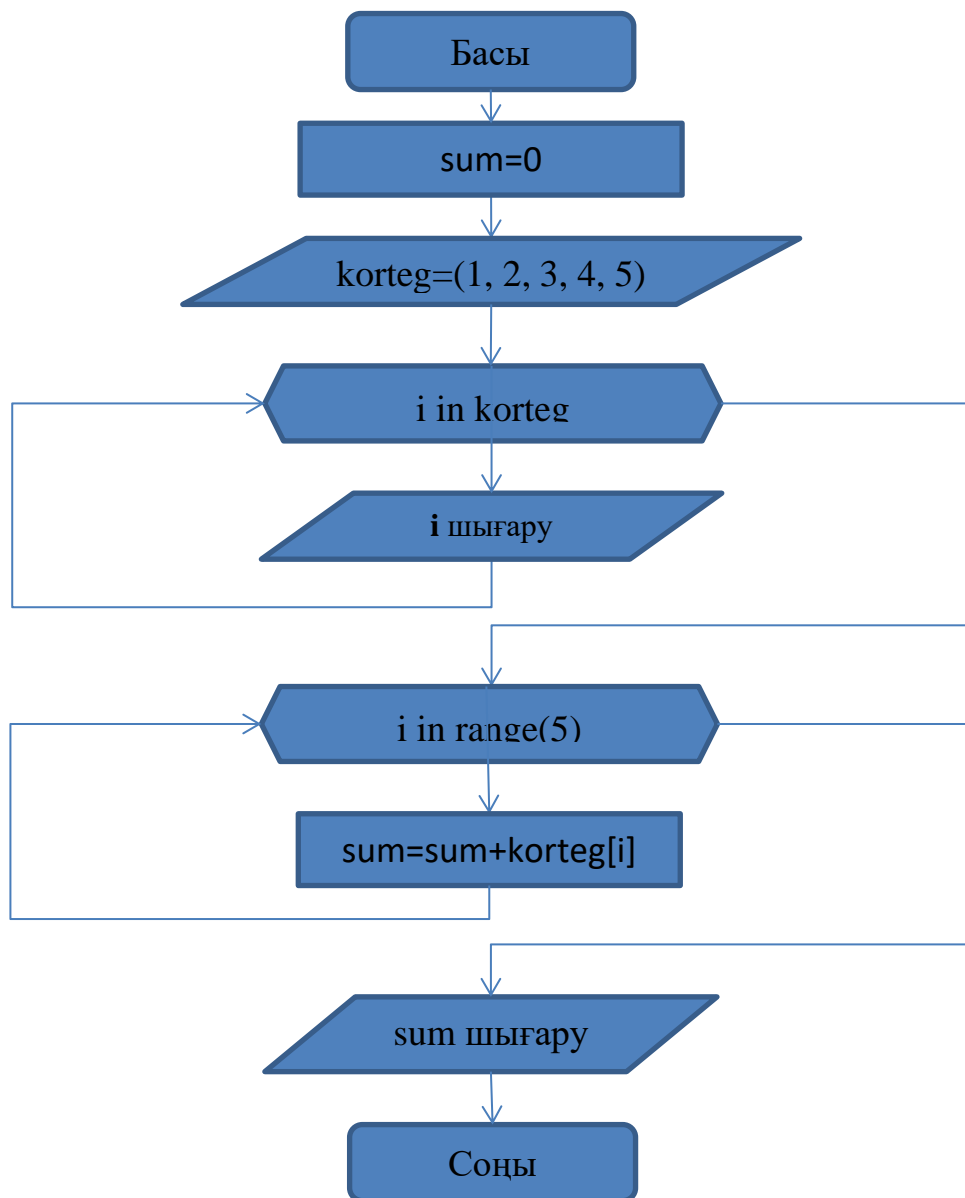
сонымен қатар, бір жолда бірнеше элементтер орналасуы мүмкін .

Бағдарламадағы кортеждің әрбір элементіне **қол жеткізу индекстің** көмегімен жүзеге асырылады - кортеж элементінің өзіндік атауы болып табылатын бүтін сан. Бағдарламада кортеж элементі туралы айтқан кезде, оның атауының артында тік жақшадағы элемент индексі болуы керек, мысалы, **korteg[i]**

Айта кету керек, кортеж элементтерін нөмірлеу бірліктен емес, нөлден басталады. Кортежді өңдеу бойынша операциялар **for** операторымен циклде элемент бойынша жүзеге асырылады.

Есеп 7.1.1. Алдын ала анықталған бүтін сандардан тұратын кортежде элементтердің қосындысын табыңыз.

Шешімі. Есепті шешу алгоритмінің блок-схемасы 72-ші суретте көрсетілген.



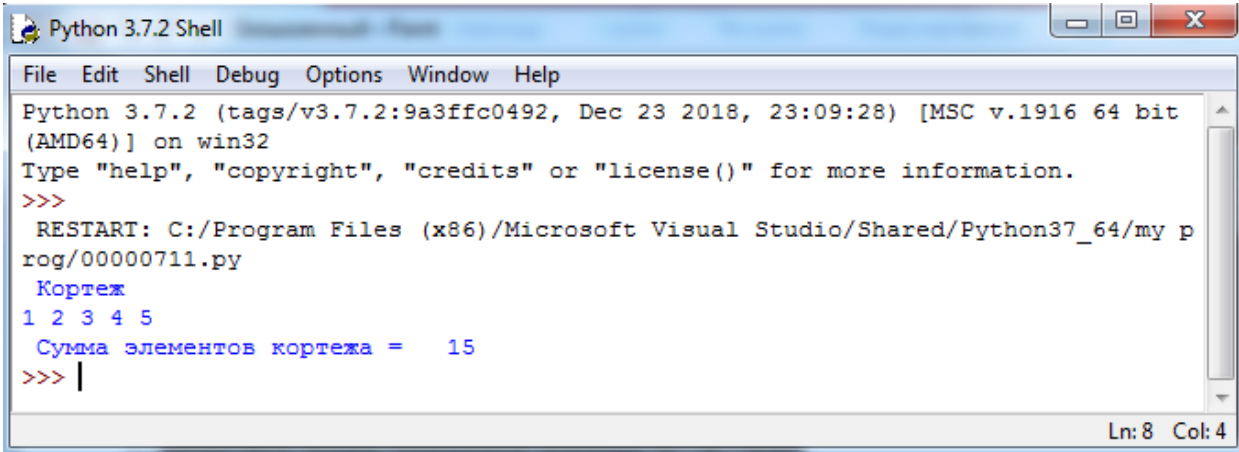
Сурет 72 – 7.1.1 есептің шешу алгоритмінің блок-схемасы

Листингте есепті шешуге арналған программа коды көрсетілген:

```
sum=0
korteg=(1, 2, 3, 4, 5)
```

```
print(" Кортеж ")
for i in korteg:
    print(i, end=" ")
for i in range(5):
    sum=sum+korteg[i]
print("\n Сумма элементов кортежа = ", sum)
```

Бағдарлама жұмысының нәтижесі 73-ші суретте 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/00000711.py
Кортеж
1 2 3 4 5
Сумма элементов кортежа = 15
>>> |
```

Сурет 73 – Кортеж элементтерінің қосындысының нәтижесі

7.2 Кортеждерді классикалық өңдеу әдістері

Деректер тізбегін өңдеумен байланысты мәселелерді шешу олардың қарапайым өңдеу әдістеріне негізделген. Тапсырманы логикалық бөліктерге бөліп, оны шешуді жеңілдетуге болады. Кортеждермен жұмыс істеудің әдеттегі міндеттері-онда берілген элементтің болуын анықтау, белгілі бір шарттарды қанағаттандыратын элементтерді таңдау және т. б.

Кортежде ұсынылған деректерді басқаруға мүмкіндік беретін барлық қолданыстағы әдістердің ішінен мыналарды бөліп көрсетуге болады:

1. Берілген жағдайда элементтер санын табу.
2. Берілген жағдайда элементтер мәндерінің қосындысын табу.
3. Берілген жағдайда элементтер мәндерінің көбейтіндісін табу.
4. Кортеж элементтерінің экстремалды мәндерін іздеу (максималды және/немесе минималды мәнді іздеу).
5. Кортеждерді біріктіру (тіркесу).
6. Кортеж ішіндегі элементтердің мәндерінің алмастыру.
7. Кортеждің бөліктері(кескінділері).

Төменде кортежді өңдеудің тізбектелген алгоритмдері келтірілген. Алғашқы төрт әдісті орындау оңай болғандықтан, сонымен қатар бұл алгоритмдер алдыңғы тақырыптардағы есепті шешу кезінде бірнеше рет қарастырылғанын ескере отырып, біз оларды Түсіндірмесіз жүзеге асырамыз. Айта кету керек, алғашқы төрт тізімде тағы бір кортеж элементі 10 санымен

салыстырылады, бірақ іс жүзінде, мысалы, **input** функциясын қолдана отырып, шарт сұрауын ұйымдастырған дұрыс.

Берілген жағдайда элементтер санын табу.

Кортеж элементтерінің **korteg=(10, 8, 29, 35, 7)** 10-нан үлкен немесе тең санын табуға мысал келтірейік. Листингте осы мысалды шешуге жауап беретін бағдарлама коды көрсетілген:

```
kol=0
korteg=(10, 8, 29, 35, 7)
print(" Кортеж ")
for i in korteg:
    print(i, end=" ")
for i in range(5):
    if korteg[i]>=10:
        kol=kol+1
print("\n Количество элементов кортежа больших или равных десяти = ",
kol)
```

Берілген жағдайда элементтер мәндерінің қосындысын табу.

Кортеж элементтерінің **korteg=(10, 8, 29, 35, 7)** 10-нан үлкен немесе тең қосындысын табуға мысал келтірейік. Листингте осы мысалды шешуге жауап беретін бағдарлама коды көрсетілген:

```
sum=0
korteg=(10, 8, 29, 35, 7)
print(" Кортеж ")
for i in korteg:
    print(i, end=" ")
for i in range(5):
    if korteg[i]>=10:
        sum=sum+korteg[i]
print("\n Сумма элементов кортежа больших или равных десяти = ", sum)
```

Берілген жағдайда элементтер мәндерінің көбейтіндісін табу.

Кортеж элементтерінің **korteg=(10, 8, 29, 35, 7)** 10-нан үлкен немесе тең көбейтіндісін табуға мысал келтірейік. Листингте осы мысалды шешуге жауап беретін бағдарлама коды көрсетілген:

```
pr=1
korteg=(10, 8, 29, 35, 7)
print(" Кортеж ")
for i in korteg:
    print(i, end=" ")
for i in range(5):
    if korteg[i]>=10:
```

```
pr=pr*korteg[i]
print("\n Произведение элементов кортежа больших или равных десяти
= ", pr)
```

Кортежде экстремалды мәндерді табу.

Біз кортеждің korteg =(10, 8, 29, 35, 7). максималды және минималды элементтерін табуға мысал келтіреміз. Листингте осы мысалды шешуге жауап беретін бағдарлама коды көрсетілген:

```
maxim=-32768
minim=32767
korteg=(10, 8, 29, 35, 7)
print(" Кортеж ")
for i in korteg:
    print(i, end=" ")
for i in range(5):
    if korteg[i]>maxim:
        maxim=korteg[i]
    if korteg[i]<minim:
        minim=korteg[i]
print("\n Максимальный элемент кортежа = ", maxim)
print("\n Минимальный элемент кортежа = ", minim)
```

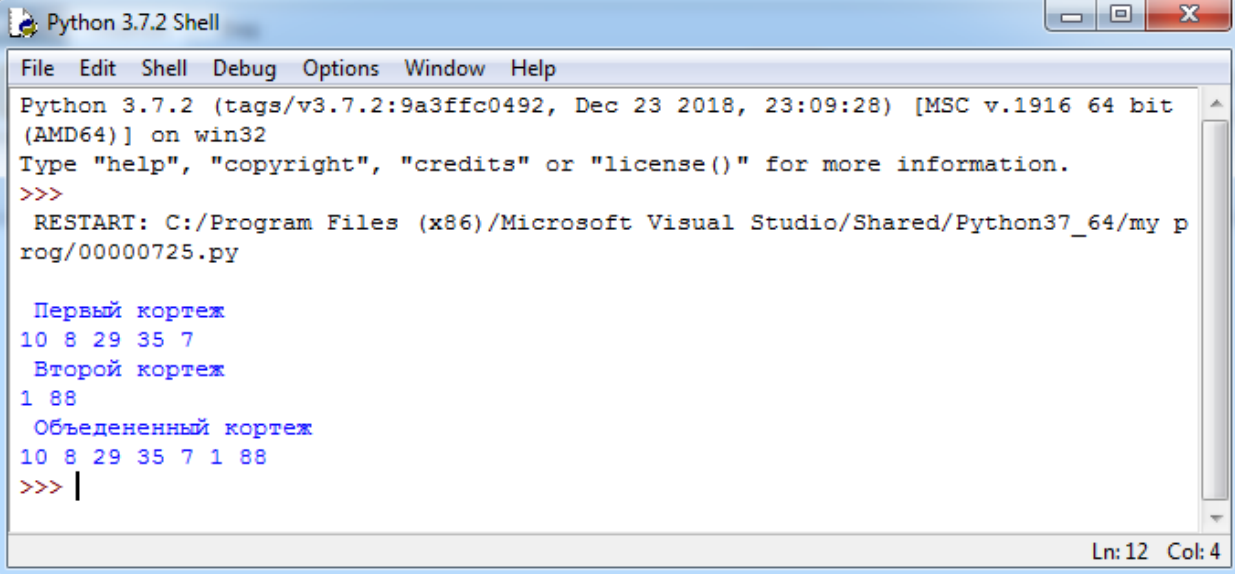
Кортеждерді біріктіру (тіркестіру).

Есеп 7.2.1. Екі кортеж берілген. Оларды өзара біріктіру керек.

Шешімі . Тапсырма алдымен екі түпнұсқа кортежді қалыптастыру және бірінші кортежге қосу (+) операциясын қолдану, содан кейін біріктірілген кортеж экранда көрсетіледі. Листингте есепті шешуге жауап беретін бағдарлама коды бар:

```
korteg1=(10, 8, 29, 35, 7)
korteg2=(1, 88)
print("\n Первый кортеж ")
for i in korteg1:
    print(i, end=" ")
print("\n Второй кортеж ")
for i in korteg2:
    print(i, end=" ")
print("\n Объединенный кортеж ")
korteg1 +=korteg2
for i in korteg1:
    print(i, end=" ")
```

Бағдарламаның нәтижесі 74-ші суретте 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/00000725.py

Первый кортеж
10 8 29 35 7
Второй кортеж
1 88
Объединенный кортеж
10 8 29 35 7 1 88
>>> |
```

Сурет 74 – 7.2.1 тапсырмасындағы екі кортеждің элементтерін біріктіру бағдарламасының нәтижесі

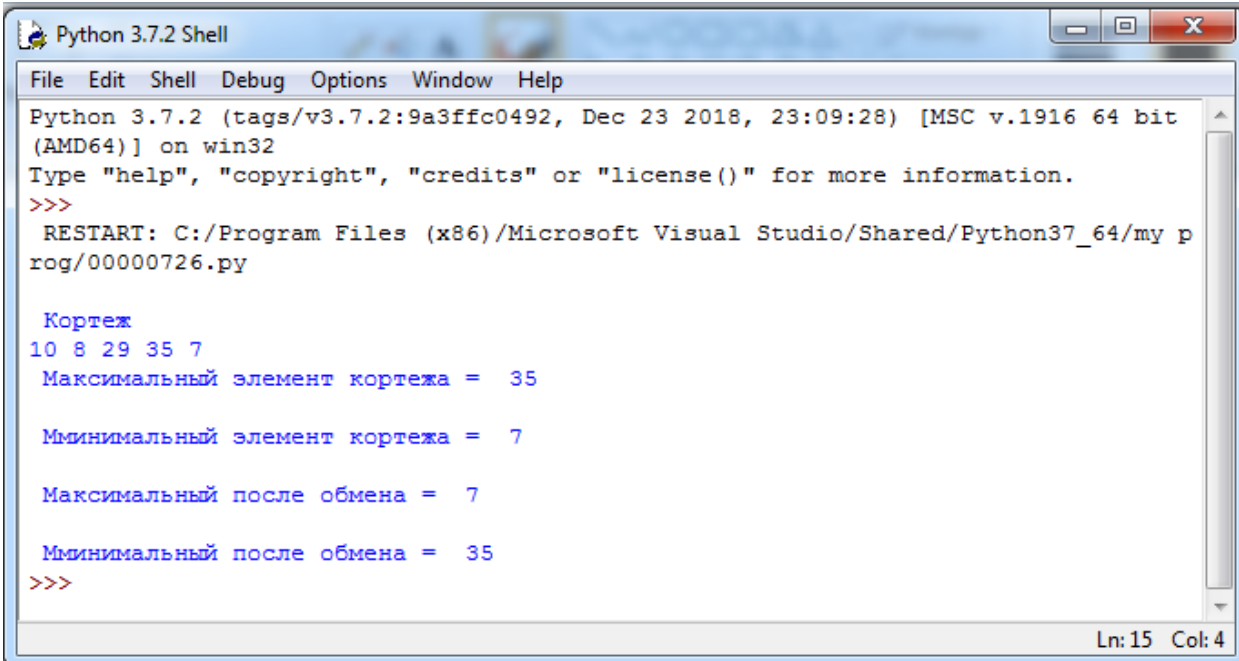
Кортеждегі элементтердің мәндерін ауыстыру.

Есеп 7.2.2. Бүгін сандар кортежіндегі максималды және минималды элементтерді тауып, оларды алмастырыңыз.

Шешім. Максималды және минималды элементтерді табу әдістері алдыңғы дәрістерде қарастырылған, сондықтан алмасудың өзі **maxim,minim=minim,maxim** операторларын орындалуынан тұратындығын ескерген жөн. Листингте есепті шешуге жауап беретін бағдарлама коды көрсетілген:

```
korteg=(10, 8, 29, 35, 7)
print("\n Кортеж ")
for i in korteg:
    print(i, end=" ")
minim=32767
maxim=-32768
for i in range(5):
    if korteg[i]>maxim:
        maxim=korteg[i]
    if korteg[i]<minim:
        minim=korteg[i]
print("\n Максимальный элемент кортежа = ", maxim)
print("\n Мминимальный элемент кортежа = ", minim)
maxim,minim=minim,maxim
print("\n Максимальный после обмена = ", maxim)
print("\n Мминимальный после обмена = ", minim)
```

Бағдарламаның нәтижесі 75-ші суретте 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/00000726.py

Кортеж
10 8 29 35 7
Максимальный элемент кортежа = 35

Минимальный элемент кортежа = 7

Максимальный после обмена = 7

Минимальный после обмена = 35
>>>
```

Сурет 75 – 7.2.2 тапсырмасындағы бағдарламасының нәтижесі

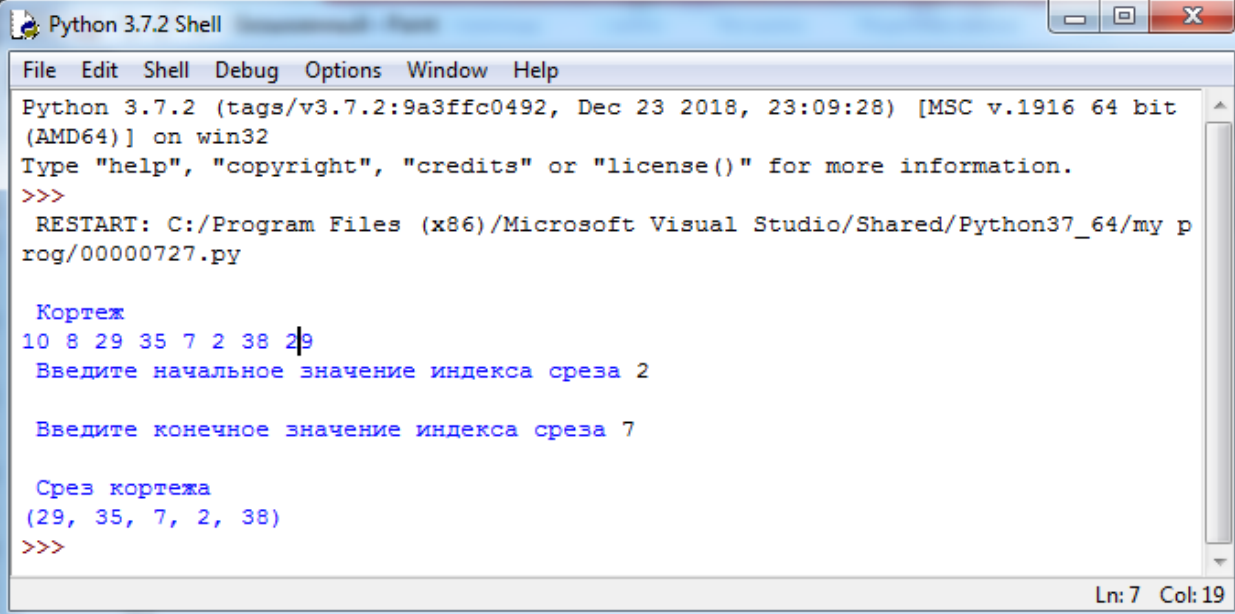
Кортеждердің кескінділері.

Кортеждің кескіндісі элементтердің алдын-ала белгіленген **a** және **b** позицияларының арасында орналасқан кортеж элементтерін шығару арқылы алынады. Кескіндер бағдарламада `print (korteg [a: b])` операторының көмегімен жүзеге асырылады.

Мысалы берілген `korteg=(10, 8, 29, 35, 7, 2, 38, 29)` кортеж кескінділері . Листингте есепті шешуге жауап беретін бағдарлама коды көрсетілген:

```
korteg=(10, 8, 29, 35, 7, 2, 38, 29)
print("\n Кортеж ")
for i in korteg:
    print(i, end=" ")
a=int(input("\n Введите начальное значение индекса среза "))
b=int(input("\n Введите конечное значение индекса среза "))
print("\n Срез кортежа ")
print(korteg[a:b])
```

Бағдарламаның нәтижесі 76-ші суретте 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/00000727.py

Кортеж
10 8 29 35 7 2 38 |
Введите начальное значение индекса среза 2

Введите конечное значение индекса среза 7

Срез кортежа
(29, 35, 7, 2, 38)
>>>
Ln: 7 Col: 19
```

Сурет 76 – Кортеж кесінділеріне арналған бағдарлама нәтижесі

7.3 Тізімдермен жұмыс істеу

Python-дағы тізімдер кортежге ұқсас жұмыс істейді айырмашылығы ауыспалы реттілікте. Осылайша, егер жаңа элементті қосу, элементті жою, сұрыптау және т.б. әрекеттерді кортежде орындалуы мүмкін болмаса, онда тізіммен жұмыс істеу әдістері бұған мүмкіндік береді.

Кортеж элементтері сияқты, тізім элементтері тек объектіге сілтеме жасайды, сондықтан тізімдерге әртүрлі типті мәліметтерді енгізуге болады. Бұл тәсіл дәстүрлі массивтерді басқа бағдарламалау тілдерінде өңдеуден ерекшеленеді, дегенмен Python-да тізімдерді өңдеу әдістерінде массивтерді өңдеу әдістерімен ұқсастық бар.

Тізімді жариялаудың синтаксисын қарастырайық .

Имя списка = [элемент 1, элемент 2, ...элемент N]

Мысалы,

```
spisok=[1, 2, 3, 4, 5]
```

Тізімдермен жұмыс істеу кезіндегі типтік тапсырмалар:

- оларда берілген элементтің болу фактісін анықтау;
- белгілі бір шарттарды қанағаттандыратын элементтерді таңдау.

Екі жағдайда да тізім элементтерін берілген үлгімен циклдік салыстыру қолданылады. Берілген үлгінің тізімде болу фактісін анықтау үшін бір сәйкестік жеткілікті, содан кейін одан әрі қарастыру тоқтатылады. Егер таңдау шарты тізімнің бірнеше элементтері үшін орындалса, онда бүкіл тізімді соңына дейін қарастыру керек.

Жоғарыда қарастырылған барлық әдістер тізімдер үшін жарамды, бірақ

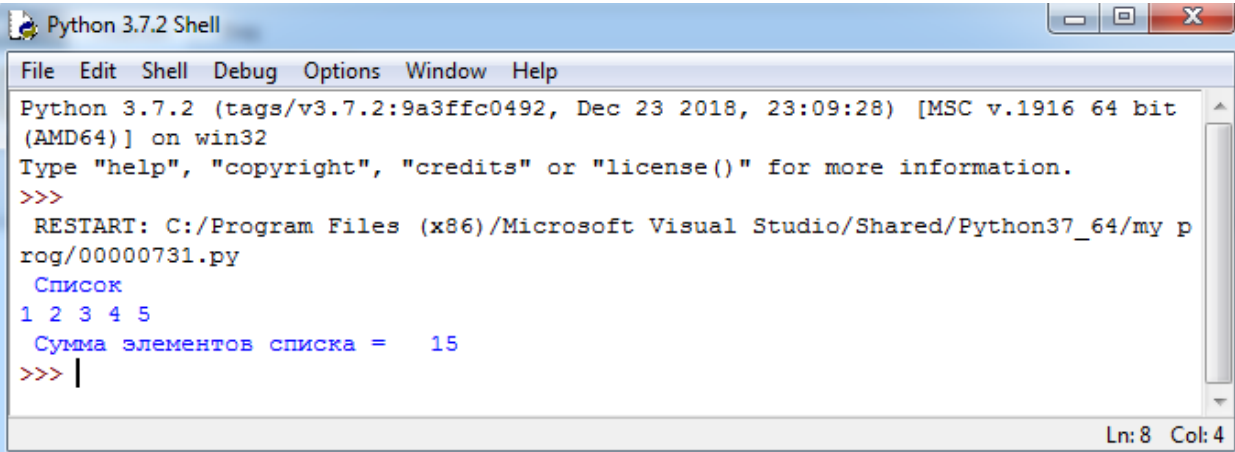
тізімдердің өзгеруіне байланысты олардың жиынтығы айтарлықтай кеңейтілуі мүмкін. Атап айтқанда, тізім элементтерін жоюға және қосуға, сұрыптауға болады.

Пайдаланушы оның элементтерін төртбұрышты жақшаға орналастыру арқылы тізім жасай алады (жоғарыда көрсетілген синтаксиске сәйкес) немесе оны кездейсоқ жасай алады. «Тізім элементтерінің қосындысын табу» тапсырмасының мысалын пайдаланып, осы әдістерді толығырақ қарастырайық. Осы тақырыптың басында біз дәл осындай есептің шешуді қарастырдық, бірақ кортеждермен жұмыс істеу үшін. Әрі қарай, Python-да тізбекті өңдеу әдістерін салыстыру пайдалы болады.

```
sum=0
spisok=[1, 2, 3, 4, 5]
print(" Список ")
for i in spisok:
    print(i, end=" ")
for i in range(5):
    sum=sum+spisok[i]
print("\n Сумма элементов списка = ", sum)
```

Тізімнен көріп отырғаныңыздай, бағдарламадағы мәтіндер мен сөйлемдер синтаксисі, іс жүзінде, өзгерген жоқ, тек пайдаланушы өңдеуге енгізген тізім элементтері тік жақшаға алынады.

Бағдарламаның нәтижесі 77-ші суретте 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/00000731.py
Список
1 2 3 4 5
Сумма элементов списка = 15
>>> |
Ln: 8 Col: 4
```

Сурет 77 – Тізім элементтерінің қосындысын табуға арналған бағдарламаның нәтижесі

Тізімдерді кездейсоқ құру мүмкіндігін қарастырайық. Оны бірнеше жолмен жасауға болады. Біріншіден, біз білетін **range** функциясында тізімнің бастапқы және соңғы мәндерін көрсетуге болады, ал **list** функциясы тізімді қайтарады. Содан кейін алдыңғы бағдарламаның коды келесі түрде болады:

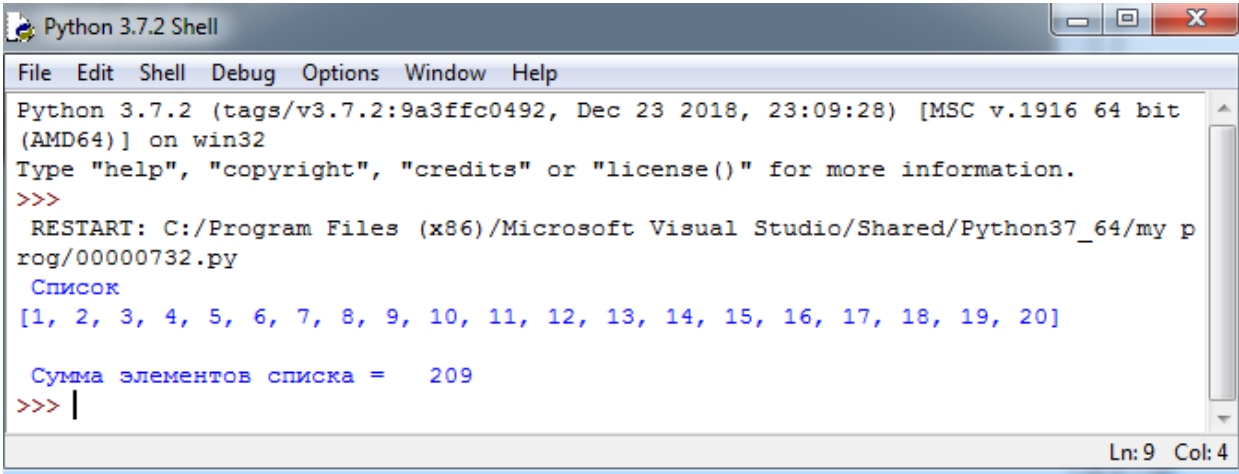
```
sum=0
```

```

spisok=list(range(1, 21))
print(" Список ")
print(spisok)
for i in range(1, 20, 1):
    sum=sum+spisok[i]
print("\n Сумма элементов списка = ", sum)

```

Бағдарлама жұмысының нәтижесі 78-ші суретте 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/00000732.py
Список
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

Сумма элементов списка = 209
>>> |
Ln: 9 Col: 4

```

Сурет 78 – List(range(1, 21) көмегімен жасалған тізім элементтерінің қосындысын табуға арналған мысал бағдарламасының нәтижесі

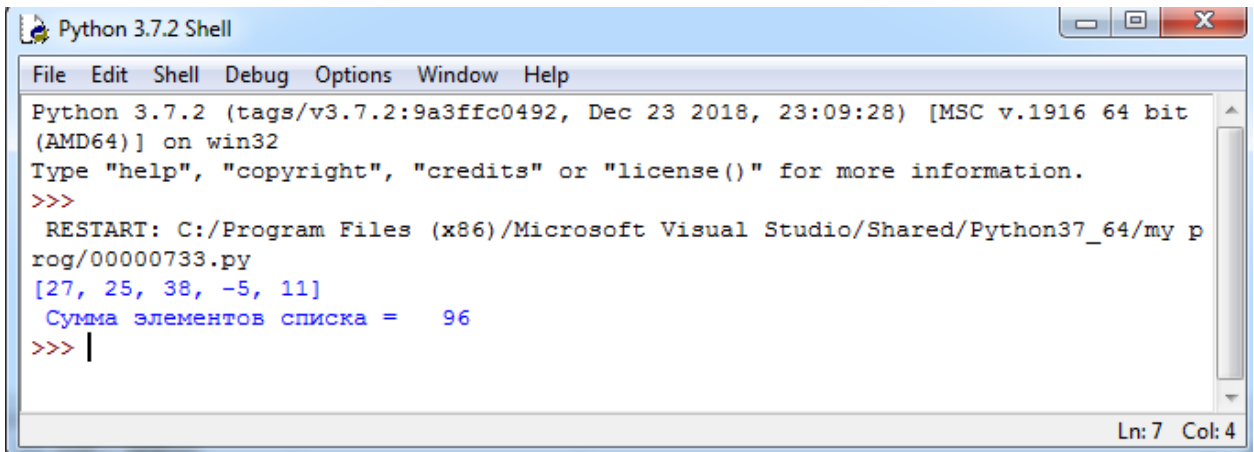
Екіншіден, сіз **sample** функциясын қолдана аласыз, ол тізім элементтерінің бастапқы тізбегінен пайдаланушы көрсеткен элементтер санын қайтарады. Бұл жағдайда **random** модулін пайдалану керек, оны **import** нұсқаулығын қолдана отырып қосу керек. Төмендегі бағдарламада көрсетілгендей, бастапқы тізім он элементтен тұрады. Содан кейін **chislo=random.sample(spisok,5)** операторымен бастапқы тізімнен бес элемент туындайды.

```

import random
sum=0
spisok=[11, 25, 38, -5, 0, 12, -78, 27, 39, 19]
chislo=random.sample(spisok, 5)
print(chislo, end=" ")
for i in range(0, 5):
    sum=sum+chislo[i]
print("\n Сумма элементов списка = ", sum)

```

Бағдарламма нәтижесі 79-ші суретте көрсетілген .



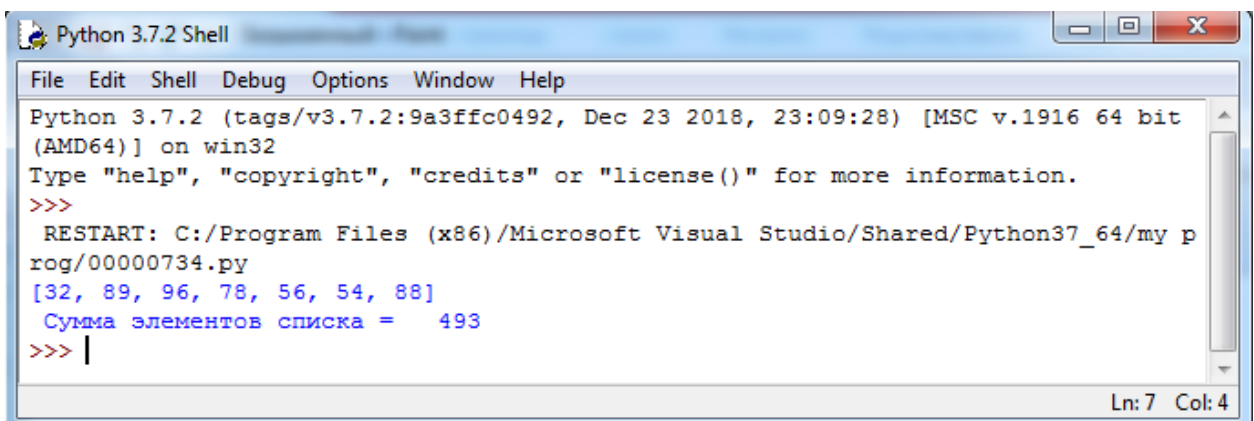
```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/00000733.py
[27, 25, 38, -5, 11]
Сумма элементов списка = 96
>>> |
```

Сурет 79- **random.sample(spisok, 5)** көмегімен жасалған тізім элементтерінің қосындысын табуға арналған мысалдың нәтижесі.

Алдыңғы кодтарды комбинациялаңыз. Алдын ала анықталған тізімнің орнына біз оны range функциясын пайдаланып өндіреміз(генерациялаймыз) , содан кейін **sample** функциясын қолданып(7 санын параметрге айналдырамыз), осылайша тізім элементтерінің санын шектейміз (жетіге дейін).

```
import random
sum=0
chislo=random.sample(range(100), 7)
print(chislo, end=" ")
for i in range(0, 7):
    sum=sum+chislo[i]
print("\n Сумма элементов списка = ", sum)
```

Бағдарлама жұмысының нәтижесі 80-ші суретте көрсетілген.



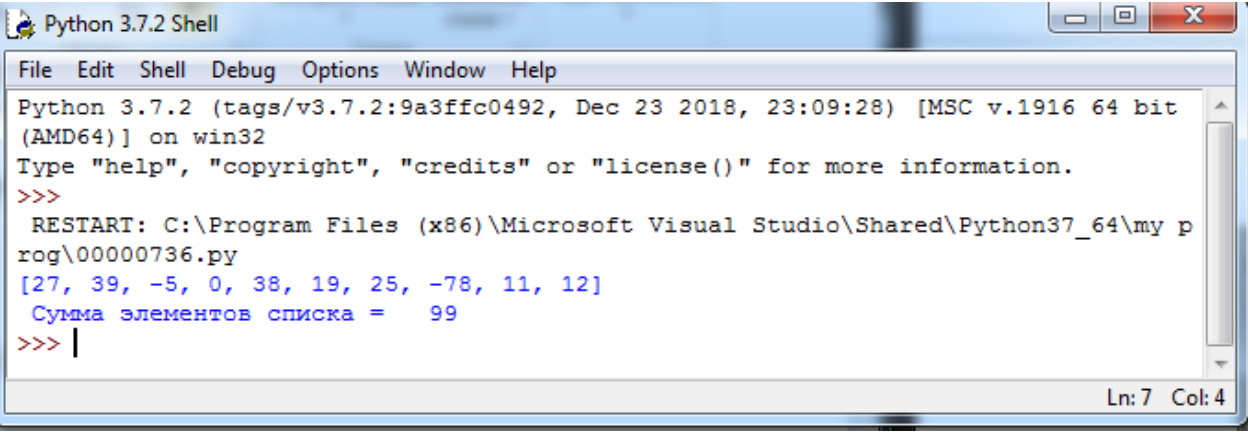
```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/00000734.py
[32, 89, 96, 78, 56, 54, 88]
Сумма элементов списка = 493
>>> |
```

Сурет 80- **0-ден 99-ға** дейінгі диапазонда құрылған 7 элементтің қосындысын табуға арналған мысалдың нәтижесі

Үшіншіден, **random** модулінде тізімді кездейсоқ араластыруға болатын **shuffle** функциясы бар.

```
import random
sum=0
spisok=[11, 25, 38, -5, 0, 12, -78, 27, 39, 19]
random.shuffle(spisok)
print(spisok, end=" ")
for i in range(0, 5):
    sum=sum+spisok[i]
print("\n Сумма элементов списка = ", sum)
```

Бағдарлама жұмысының нәтижесі 81-ші суретте 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\00000736.py
[27, 39, -5, 0, 38, 19, 25, -78, 11, 12]
Сумма элементов списка = 99
>>> |
```

Сурет 81- Бастапқы тізімнен алынған 5 элементтің қосындысын табуға арналған мысал бағдарламасының нәтижесі

Бағдарламада тізімдерді құрудың қарастырылған тәсілдерінен басқа, сіз динамикалық тізімді құру әдісін қолдана аласыз. Төмендегі кодта көрсетілгендей, бос тізімді алдымен **sp = []** операторы жариялайды. Содан кейін пайдаланушыдан тізім элементтерінің саны сұралады. Циклде пайдаланушы тізім элементтерін енгізе бастайды, ал **append()** әдісі оларды тізімге қосуға мүмкіндік береді.

```
sp=[]
n=int(input("\n Введите количество элементов списка "))
for i in range(n):
    chislo=int(input("\n Введите число "))
    sp.append(chislo) # Добавляем элементы списка
for i in range(n): # Выводим элемента списка
    print(sp[i], end=" ")
```

Бағдарлама жұмысының нәтижесі 82-ші суретте 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/00000738.py

Введите количество элементов списка 5

Введите число 8

Введите число 9

Введите число 1

Введите число 0

Введите число 6
8 9 1 0 6
>>>
Ln: 18 Col: 4

```

Сурет 82 –Тізімді «қолмен» құру бағдарламасының нәтижесі»

Осылайша, біз Python-да тізімдер құрудың негізгі тәсілдерін қарастырдық. Тізімдерді құрудың жоғарыда қарастырылған негізгі әдістерін біріктіру арқылы тізімдерді орнатуға болады. Әрі қарай осы әдістердің кейбіреулері есептерді шешуде қолданылады.

Python-да тізімдерімен жұмыс істеу үшін әдістер қарастырылған, олардың кейбіреулері 6-ші кестеде келтірілген. Олардың қолданылуын мысалдармен қарастырайық.

Кесте 6 – Тізімдер жұмыс істеу әдістері

Әдіс	Әдістің сипаттамасы
<code>spisok.append(x)</code>	Тізімнің соңына <code>x</code> мәнін қосады
<code>spisok.insert(i, x)</code>	<code>i</code> позициясына <code>x</code> мәнін қосады
<code>spisok.extend(spisok1)</code>	<code>Spisok1</code> тізімінің барлық элементтерін соңына қосу арқылы тізімді кеңейтеді
<code>spisok.remove(x)</code>	<code>x</code> мәні бар тізімдегі бірінші элементті жояды
<code>spisok.pop(i)</code>	<code>i</code> нөмірінің мәнін позицияға қайтарады және сонымен бірге оны тізімнен жояды. Егер аргумент берілмесе, тізімнің соңғы элементі қайтарылады және жойылады
<code>spisok.count(x)</code>	<code>x</code> мәні бар элементтер санын қайтарады
<code>spisok.sort([reverse = True])</code>	Элементтерді өсу бойынша сұрыптайды. Reverse параметрі міндетті емес және логикалық мәндерді қабылдайды. Егер сіз True мәнін берсеңіз, тізім кему бойынша сұрыпталады
<code>spisok.reverse()</code>	Кері тізімді қайтарады

Есеп 7.3.1. Кездейсоқ құрылған тізімге пайдаланушы енгізген элементті қосыңыз

Шешімі. Төмендегі бағдарламадан көріп отырғаныңыздай, алдымен тізімнің генерациясы жасалады, содан кейін сан сұралады және **append** әдісін қолдана отырып, тізімнің соңына сан қосылады.

```
import random
spisok=random.sample(range(100), 7)
print(spisok, end=" ")
chislo=int(input("\n Введите число "))
spisok.append(chislo) # Добавляем введенное число в конец списка
print(spisok, end=" ")
```

Бағдарлама жұмысының нәтижесі 83-ші суретте 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/000007311.py
[25, 32, 28, 23, 89, 59, 0]
Введите число 39
[25, 32, 28, 23, 89, 59, 0, 39]
>>>
```

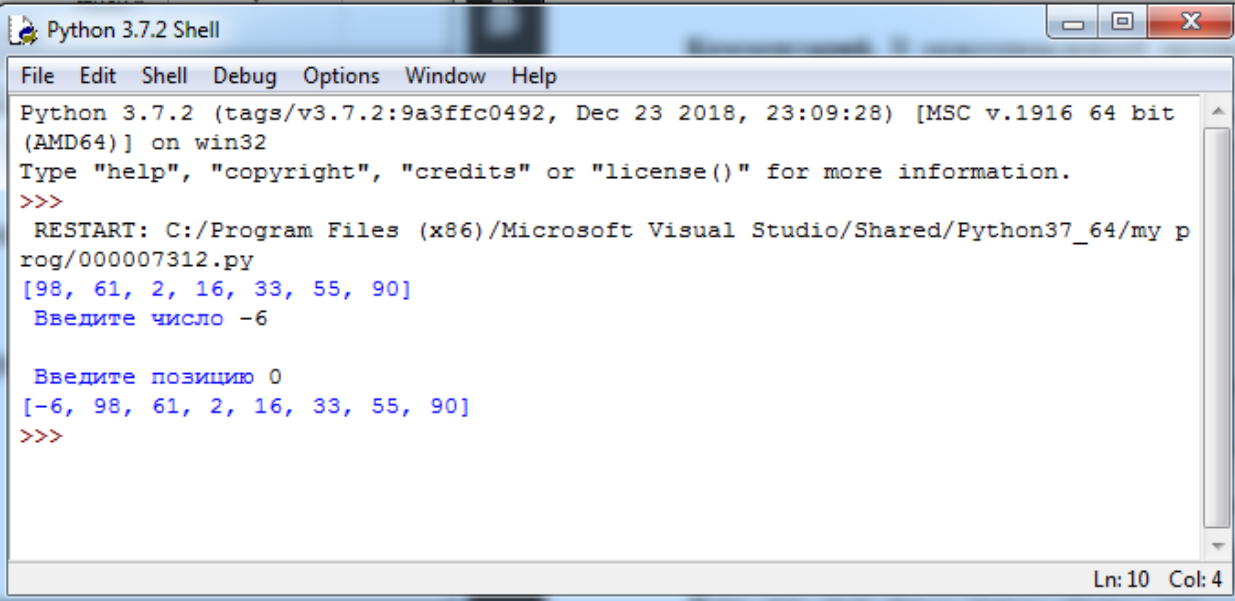
Сурет 83 – Тізімнің соңына санды қосу бағдарламасының нәтижесі

Есеп 7.3.2. Кездейсоқ құрылған тізімге пайдаланушы енгізген элементті көрсетілген позицияға қосыңыз.

Шешімі. Төмендегі бағдарламада алдымен тізім жасалады, содан кейін, санды қосуға арналған , сан мен позиция нөмірі сұралады. Содан кейін **insert()** әдісін қолдана отырып тиісті **poz ,chislo** параметрлері бар, санды көрсетілген позицияға енгізеді.

```
import random
spisok=random.sample(range(100), 7)
print(spisok, end=" ")
chislo=int(input("\n Введите число ")) # Вводим число
poz=int(input("\n Введите позицию ")) # Вводим позицию для добавления
введенного выше числа в список
spisok.insert(poz, chislo) # Добавляем введенное число
print(spisok, end=" ")
```

Бағдарлама жұмысының нәтижесі 84-ші суретте 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/000007312.py
[98, 61, 2, 16, 33, 55, 90]
Введите число -6

Введите позицию 0
[-6, 98, 61, 2, 16, 33, 55, 90]
>>>
Ln: 10 Col: 4
```

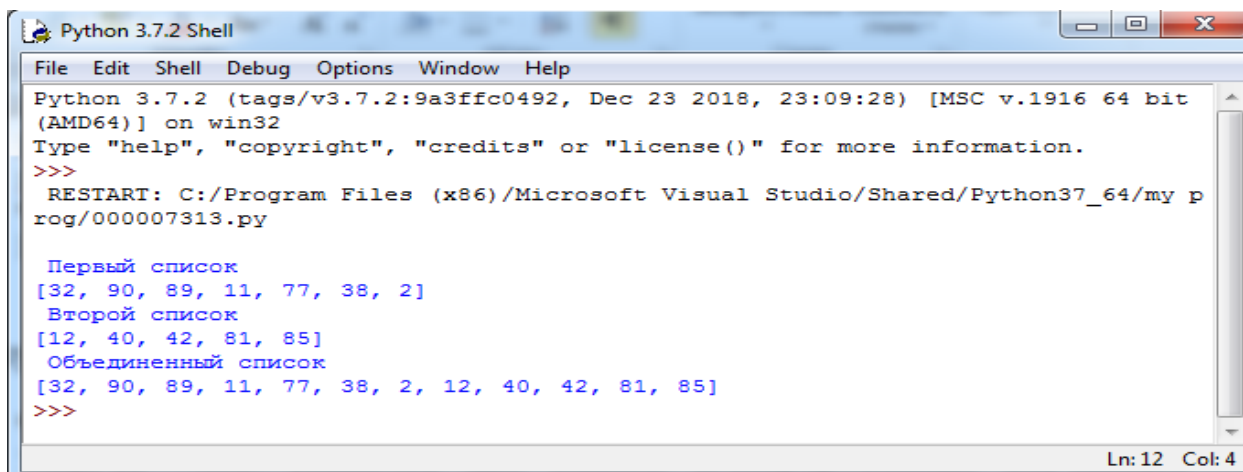
Сурет 84 – Енгiзiлген санды тiзiмдегi енгiзiлген позицияға қосу бағдарламасының нәтижесi

Есеп 7.3.3. Кездейсоқ құрылған екі тізім бар. Бірінші тізімнің соңына екінші тізімнің барлық элементтерін қосыңыз.

Шешiмi . Алдымен **spisok1** және **spisok2** кездейсоқ құрылған екі тізімді іске асырамыз. Бірінші тізімге **extend ()** әдісін қолдану арқылы біз екінші тізімнің элементтерімен оны кеңейтеміз.

```
import random
spisok1=random.sample(range(100), 7)
print("\n Первый список ")
print(spisok1, end=" ")
spisok2=random.sample(range(100), 5)
print("\n Второй список ")
print(spisok2, end=" ")
spisok1.extend(spisok2) # Добавляем к первому списку элементы второго
списка
print("\n Объединенный список ")
print(spisok1, end=" ")
```

Бағдарлама жұмысының нәтижесі 85-ші суретте 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/000007313.py

Первый список
[32, 90, 89, 11, 77, 38, 2]
Второй список
[12, 40, 42, 81, 85]
Объединенный список
[32, 90, 89, 11, 77, 38, 2, 12, 40, 42, 81, 85]
>>>
Ln: 12 Col: 4
```

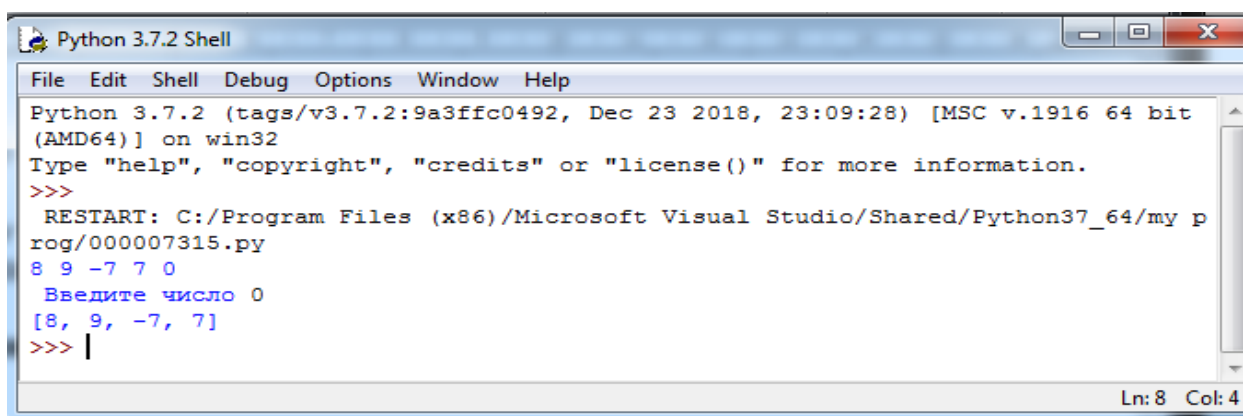
Сурет 85 – Басқа тізімнен тізімге қосу бағдарламасының нәтижесі

Есеп 7.3.4. Алдын ала құрылған тізімнен пайдаланушы енгізген элементті жою керек

Шешімі. Бастапқы тізімде бес элемент болады. Пайдаланушы жойылатын элементтің мәнін енгізеді. Тізімге қолданылған `remove()` әдісі көрсетілген әрекеттерді жүзеге асырады.

```
spisok=[8, 9, -7, 7, 0]
for i in spisok:
    print(i, end=" ")
chislo=int(input("\n Введите число "))
spisok.remove(chislo)
print(spisok, end=" ")
```

Бағдарлама жұмысының нәтижесі 86-ші суретте 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/000007315.py
8 9 -7 7 0
Введите число 0
[8, 9, -7, 7]
>>> |
Ln: 8 Col: 4
```

Сурет 86 – Енгізілген санды тізімінен жою бойынша бағдарламаның нәтижесі

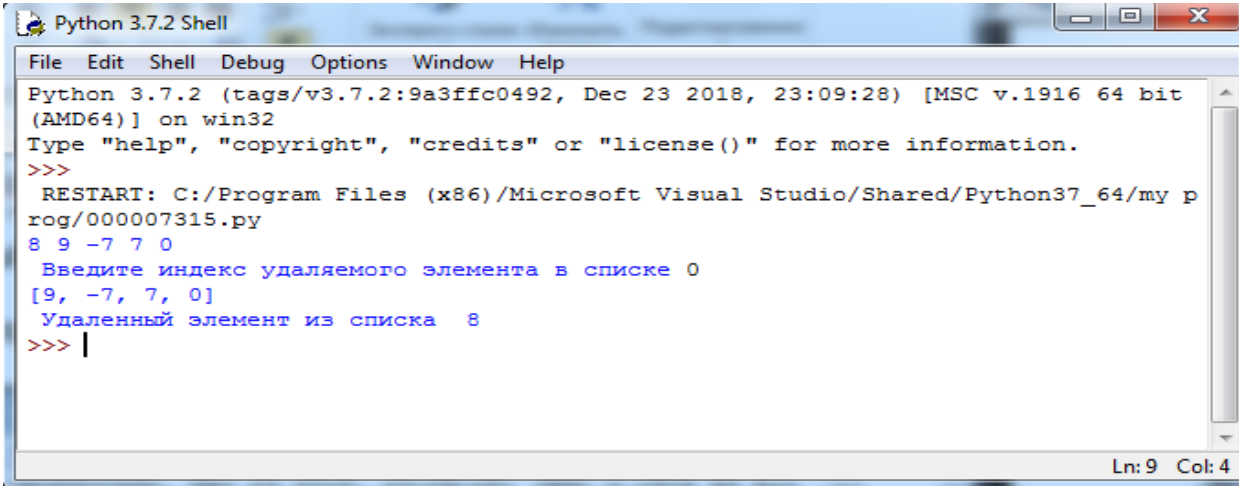
Есеп 7.3.5. Бастапқы тізімнен пайдаланушы көрсеткен позициядан элементті жою керек.

Шешімі . Бастапқы тізім бес элементтен тұрады. Пайдаланушы жойылатын элементтің индексін енгізеді. Егер аргумент `pop()` әдісіне

берілмесе, онда тізімнің соңғы элементі қайтарылады және жойылады. Алайда, **pop ()** әдісінің параметрі ретінде біз элементтің индексін көрсетеміз. Экранға нәтижелік тізімнен басқа, жойылған элементтің мәнін шығарамыз .

```
spisok=[8, 9, -7, 7, 0]
for i in spisok:
    print(i, end=" ")
ind=int(input("\n Введите индекс удаляемого элемента в списке "))
chislo=spisok.pop(ind)
print(spisok, end=" ")
print("\n Удаленный элемент из списка ", chislo)
```

Бағдарлама жұмысының нәтижесі 87-ші суретте 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/000007315.py
8 9 -7 7 0
Введите индекс удаляемого элемента в списке 0
[9, -7, 7, 0]
Удаленный элемент из списка 8
>>> |
```

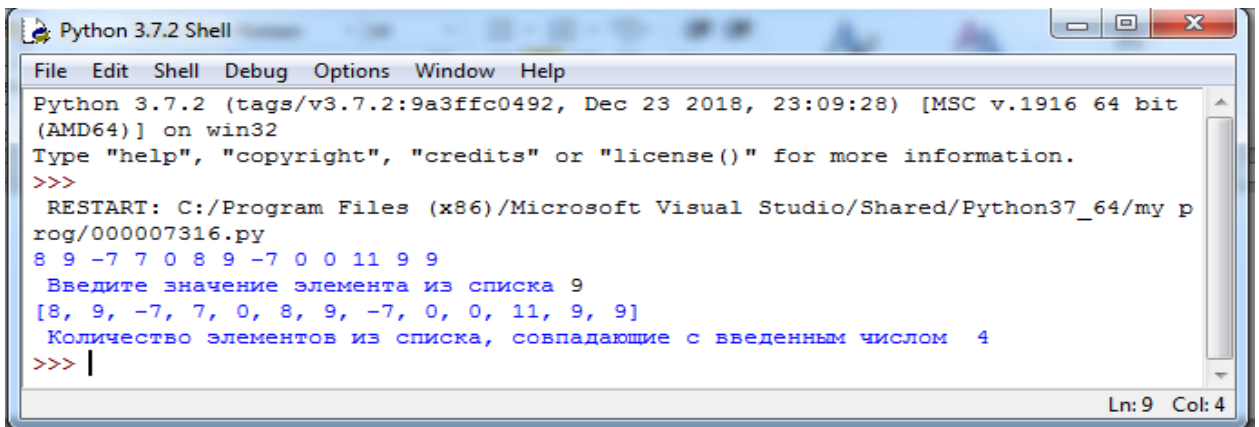
Сурет 87 – позиция бойынша тізімнен санды жою бағдарламаның нәтижесі

Есеп 7.3.6. Бастапқы тізімде белгілі бір мәні бар элементтер санын есептеңіз.

Шешімі . Біз тізімді бірнеше қайталанатын элементтерді кірістіру арқылы құрастырдық. Пайдаланушы енгізген **znach** мәні- **count ()** әдісінің параметрі , ол **znach** элементінің қайталану санын қайтарады.

```
spisok=[8, 9, -7, 7, 0, 8, 9, -7, 0, 0, 11, 9, 9]
for i in spisok:
    print(i, end=" ")
znach=int(input("\n Введите значение элемента из списка "))
kol=spisok.count(znach)
print(spisok, end=" ")
print("\n Количество элементов из списка, совпадающие с введенным
числом ", kol)
```

Бағдарлама жұмысының нәтижесі 88-ші суретте 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/000007316.py
8 9 -7 7 0 8 9 -7 0 0 11 9 9
Введите значение элемента из списка 9
[8, 9, -7, 7, 0, 8, 9, -7, 0, 0, 11, 9, 9]
Количество элементов из списка, совпадающие с введенным числом 4
>>> |
```

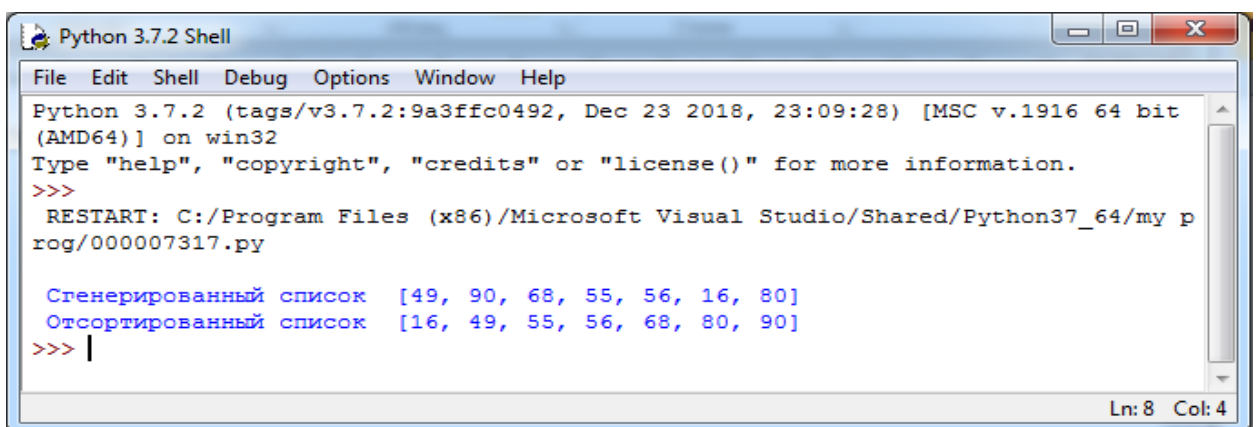
Сурет 88 – Енгiзiлген санның тiзiмiнде қайталануларды табу бағдарламасының нәтижесi

Есеп 7.3.7. Кездейсоқ құрылған тiзiмдi өсу бойынша сұрыптау керек.

Шешiмi . Python-да тiзiмдi сұрыптау үшiн мiндеттi емес **reverse** параметрi бар **sort ()** әдiсiн қолданады. Төмендегi кодта ол **False** жағдайда орнатылған , бұл тiзiмдi өсу бойынша сұрыптауға мүмкiндiк бередi.

```
import random
spisok=random.sample(range(100), 7)
print("\n Сгенерированный список ", spisok, end=" ")
spisok.sort(reverse=False)
print("\n Отсортированный список ", spisok, end=" ")
```

Бағдарлама жұмысының нәтижесi 89-шi суретте көрсетiлген



```
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/000007317.py

Сгенерированный список [49, 90, 68, 55, 56, 16, 80]
Отсортированный список [16, 49, 55, 56, 68, 80, 90]
>>> |
```

Сурет 89 – Тiзiмдi өсу бойынша сұрыптау бағдарламасының нәтижесi

Есеп 7.3.8. Кездейсоқ түрде жасалған тiзiмдi керi ретпен көрсетiңiз.

Шешiмi. Кездейсоқ құрылған бастапқы тiзiмге қолданылған параметрлерсiз **reverse** әдiсi керi тiзiмдi қайтарады.

```
import random
```

```

spisok=random.sample(range(100), 7)
print("\n Сгенерированный список ", spisok, end=" ")
spisok.reverse()
print("\n Список в обратном порядке ", spisok, end=" ")

```

Бағдарлама жұмысының нәтижесі 90-ші суретте 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/000007318.py

Сгенерированный список [90, 87, 0, 81, 35, 22, 57]
Список в обратном порядке [57, 22, 35, 81, 0, 87, 90]
>>> |
Ln: 8 Col: 4

```

Сурет 90 – Кері ретімен тізімді шығару бағдарламасының нәтижесі

7.4 Сөздіктермен жұмыс істеу

Python-да элементтердің нөмірлері(индекстері) ғана емес, сонымен қатар жол мәндері бар деректер құрылымдарымен жұмыс істеу мүмкіндігі бар. Көбінесе бағдарламашы әріптік-сандық белгілері бар мәліметтермен жұмыс істеуге мәжбүр болғандықтан, бұл өте ыңғайлы. Мұндай деректерді өңдеу, мысалы, жүктер мен жолаушыларды тасымалдау компанияларында қолданылатын ақпараттық жүйелерде пайдаланылуы мүмкін.

Python тілінде оның элементтерін сандық индекспен емес, еркін объектімен(сан, жол, кортеж) анықтауға мүмкіндік беретін мәліметтер құрылымы **сөздік** деп аталады. Белгілі бір ерікті объект-бұл сөздіктегі объектілер жиынтығына қол жеткізуге мүмкіндік беретін **кілт**.

Сөздікті құру синтаксисі әртүрлі болуы мүмкін, біз олардың біреуін береміз:

```

Сөздік аты   = {кілт:мән,
                кілт:мән,
                кілт:мән,
                кілт:мән }

```

Көріп отырғаныңыздай, сөздіктің әр элементі "**кілт:мән**" жұбынан тұрады. Кілт сөздік элементін анықтайды, ал мағынасы осы кілтке сәйкес келетін мәліметтер. Әр жұп бағдарламаларда әр түрлі жолдарда жазылуы мүмкін немесе бір жолда орналасуы мүмкін.

Бұл құрылымның мағынасы C#, Microsoft Visual Basic, Pascal және т.б. сияқты танымал бағдарламалау тілдерінде жұмыс істейтін таңдау операторын (Case, Select Case, switch) қолдануға ұқсайды, бірақ Python-да сөздіктерді қолдану мүмкіндігі әлдеқайда кең. Сөздіктерді объектілер арасында сәйкестік орнату, объектілер санын есептеу, объектіге қатысты деректерді сақтау қажет болған кезде пайдалануға болады.

Сөздіктерді пайдалану кезінде келесі ережелерді ескеру қажет:

- 1."Кілт:мән" жұпындағы екі бірдей кілт болуы мүмкін емес.
- 2.Сөздіктер тізбектікке жатпайды. Осылайша, тізімдер мен кортеждерде жұмыс істеу үшін пайдаланылған функциялар сөздіктер үшін қолайсыз.
3. Кілт өзгермейтін болуы керек, ол бүтін немесе нақты сан, жол, кортеж болуы мүмкін. Бұл бірінші тармақта талаптың орындалуына қол жеткізу үшін жасалады.
4. Мәндер кез-келген деректер түрін қабылдай алады, өзгертілетін және өзгермейтін болады.

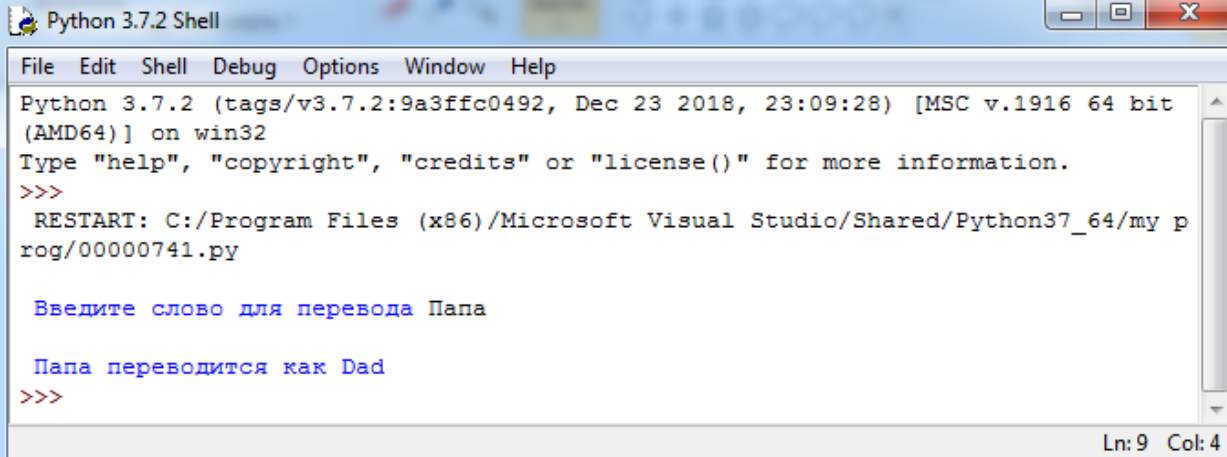
Сөздік көмегімен бағдарламалардың бірнеше мысалын қарастырыңыз.

Есеп 7.4.1. Орыс тілінің бірнеше сөзін ағылшын тіліне аударатын бағдарламаны құрыңыз.

Шешімі. Бұл бағдарламада сөздік жасау жоғарыда айтылғаннан өзгеше. Алдымен **slovar=dict()** операторымен **slovar** деп аталатын бос сөздік құрамыз. Әрі қарай, біз сөздікте кілттерден(орыс сөздері) және мағыналардан (ағылшын тіліндегі сөздер) тұратын жұптар құрамыз. Пайдаланушы ағылшын тіліне аударғысы келетін сөзді енгізгенде, сөздік элементтерін санау ұйымдастырылады, ал егер олардың біреуі енгізілген сөзбен сәйкес келсе, жауап көрсетіледі.

```
slovar=dict()
slovar['Мама']='Mother'
slovar['Папа']='Dad'
slovar['Бабушка']='Grandmother'
slovar['Дедушка']='Grandfather'
slovo=input("\n Введите слово для перевода ")
if slovo in slovar:
    perevod=slovar[slovo]
    print("\n", slovo, "переводится как", perevod)
```

Бағдарлама жұмысының нәтижесі 91-ші суретте 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/00000741.py

Введите слово для перевода Папа

Папа переводится как Dad
>>>
Ln: 9 Col: 4
```

Сурет 91 –Сөздерді сөздіктен таңдау нәтижесі

Бағдарламаны сөздікке жаңа сөзді қосу мүмкіндігімен толықтырайық. Мұны істеу үшін, сөзді сөздіктен енгізбеген жағдайда, біз сөзді енгізуді және осы сөзді сөздікке аударуды ұйымдастырамыз. Сөздікте енгізілген сөздің бар екеніне көз жеткізу үшін сөздікті тексеруді ұйымдастырамыз.

```
slovar=dict()
slovar['Мама']='Mother'
slovar['Папа']='Dad'
slovar['Бабушка']='Grandmother'
slovar['Дедушка']='Grandfather'
slovo=input("\n Введите слово для перевода ")
if slovo in slovar:
    perevod=slovar[slovo]
    print("\n", slovo, "переводится как", perevod)
slovo=input("\n Какое слово для перевода добавляем? ")
if slovo not in slovar:
    perevod=input("\n Введите перевод слова ")
    slovar[slovo]=perevod
    print("\n Слово дабавлено в словарь")
slovo=input("\n Проверка словаря на наличие добавленного слова ")
if slovo in slovar:
    perevod=slovar[slovo]
    print("\n", slovo, "переводится как", perevod)
```

Бағдарлама жұмысының нәтижесі 92-ші суретте 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/00000742.py

Введите слово для перевода Мышь

Какое слово для перевода добавляем? Мышь

Введите перевод слова Mouse

Слово дабавлено в словарь

Проверка словаря на наличие добавленного слова Мышь

Мышь переводится как Mouse
>>>
Ln: 17 Col: 4

```

Сурет 92 – Сөзді сөздікке қосу нәтижесі

Есеп 7.4.2. Шет тілдерінің бірінде "Сәлем, бағдарламашы" фразасын шығаратын бағдарламаны құрыңыз.

Шешімі. Бұл есепті шешу кезде біз кілт - ел және сәлемдесу мағынасы бар бірнеше жұптан сөздік құраймыз. Барлық жұптарды фигуралық жақшаға аламыз. Әрі қарай, шартты **if** операторын қолдана отырып, енгізілген ел атауы сөздіктегі кілтке сәйкес келетіндігін тексереміз, егер шарт ақиқат болса, біз тиісті мәнді шығарамыз.

```

slovar={"Россия":"Привет, программист!",
        "Англия":"Hello, programmer!",
        "Германия":"Hallo, Programmierer!",
        "Испания":"Hola, programador!",
        "Италия":"Ciao, programmatore!"}
slovo=input("\n Введите название страны ")
if slovo in slovar:
    perevod=slovar[slovo]
    print("\n", "Приветствие страны", slovo, "пишется как", perevod)

```

Бағдарлама жұмысының нәтижесі 93-ші суретте 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/00000743.py

Введите название страны Испания

Приветствие страны Испания пишется как Hola, programador!
>>> |
Ln: 9 Col: 4

```

Сурет 93 – Сәлемдесу сөздігінен елді таңдау нәтижесі

Есеп 7.4.3. Кинотеатрға билетке тапсырыс беруге ұқсайтын бағдарламаны жасаңыз. Оның нәтижесі келесі ақпараттан тұрады : таңдалған фильм, сеанс уақыты, зал, қатар нөмірі мен орны.

Шешімі . Үш сөздікті құрастырамыз. Олардың біріншісінде **film** атауы бар, сөздік элементтері фильмнің атауына және сеанс уақытына сәйкес келеді. Уақытқа байланысты сеанс белгілі бір залда өтеді деп болжанады, сондықтан екінші сөздікте (**time**) элементтері зал нөмірі мен сеанстың басталу уақыты болады. Үшінші сөздік (**place**) қатардың нөмірі мен орнын сақтау үшін қажет.

Фильмнің атауын **if name in film** шартымен енгізгеннен кейін біз сөздікте фильмнің болуын тексереміз, егер ол бар болса, сұралған фильм үшін қол жетімді сеанс уақытын көрсетеміз. Содан кейін біз сеанс уақытын сұраймыз, Егер ол сөздікте (**if vrem in time**) көрсетілсе, онда берілген уақытқа қатысты тиісті зал таңдалады, содан кейін пайдаланушыдан қатар нөмірі мен орнын енгізу сұралады. Бағдарлама коды төмендегі тізімде көрсетілген.

```
film={"Мавританец":"16-00 18-00 20-00 22-00",
      "Конь Юлий и большие скачки":"9-00 11-00 13-00",
      "Лига справедливости Зака Снайдера":"12-00 14-15 16-30 19-00"}
time={"9-00":"Зал 1",
      "11-00":"Зал 1",
      "12-00":"Зал 2",
      "13-00":"Зал 1",
      "14-15":"Зал 3",
      "16-00":"Зал 1",
      "16-30":"Зал 2",
      "18-00":"Зал 3",
      "19-00":"Зал 1",
      "20-00":"Зал 2",
      "22-00":"Зал 3"}
place={"1":"1-20",
       "2":"1-20",
       "3":"1-20",
       "4":"1-20",
       "5":"1-20",
       "6":"1-20",
       "7":"1-20",
       "8":"1-20",
       "9":"1-20",
       "10":"1-20",
       "11":"1-20",
       "12":"1-20"}
```

```
print("Доступные фильмы: Конь Юлий и большие скачки, Лига справедливости Зака Снайдера, Мавританец")
```

```
name=input("\n Введите название фильма: ")
```

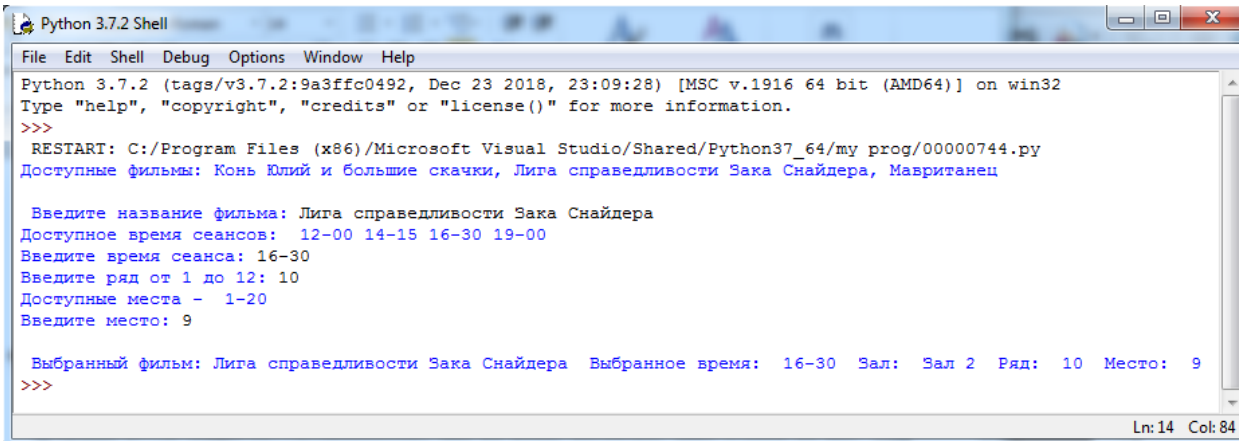
```
if name in film:
```

```

print("Доступное время сеансов: ", film[name])
vrem=input("Введите время сеанса: ")
if vrem in time:
    zal=time[vrem]
    if zal=="Зал 1":
        ryad=input("Введите ряд от 1 до 12: ")
        if ryad in place:
            print("Доступные места - ", place[ryad])
            mesto=int(input("Введите место: "))
            if (mesto>0) and (mesto<21):
                print("\n Выбранный фильм:", name, " Выбранное время: ",
vrem, " Зал: ", zal," Ряд: ", ryad, " Место: ", mesto)
            else:
                print("Введено некорректное место")
        else:
            print("Введен некорректный номер ряда")
    elif zal=="Зал 2":
        ryad=input("Введите ряд от 1 до 12: ")
        if ryad in place:
            print("Доступные места - ", place[ryad])
            mesto=int(input("Введите место: "))
            if (mesto>0) and (mesto<21):
                print("\n Выбранный фильм:", name, " Выбранное время: ",
vrem, " Зал: ", zal," Ряд: ", ryad, " Место: ", mesto)
            else:
                print("Введено некорректное место")
        else:
            print("Введен некорректный номер ряда")
    elif zal=="Зал 3":
        ryad=input("Введите ряд от 1 до 12: ")
        if ryad in place:
            print("Доступные места - ", place[ryad])
            mesto=int(input("Введите место: "))
            if (mesto>0) and (mesto<21):
                print("\n Выбранный фильм:", name, " Выбранное время: ",
vrem, " Зал: ", zal," Ряд: ", ryad, " Место: ", mesto)
            else:
                print("Введено некорректное место")
        else:
            print("Введен некорректный номер ряда")
    else:
        print("Неверно введено время сеанса")
else:
    print("Неверное введено название фильма")

```


Бағдарлама жұмысының нәтижесі 94-ші суретте 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 prog/00000744.py
Доступные фильмы: Конь Юлий и большие скачки, Лига справедливости Зака Снайдера, Мавританец

Введите название фильма: Лига справедливости Зака Снайдера
Доступное время сеансов: 12-00 14-15 16-30 19-00
Введите время сеанса: 16-30
Введите ряд от 1 до 12: 10
Доступные места - 1-20
Введите место: 9

Выбранный фильм: Лига справедливости Зака Снайдера Выбранное время: 16-30 Зал: Зал 2 Ряд: 10 Место: 9
>>>
```

Сурет 94 – Кинотеатр кассасының жұмысына ұқсайтын бағдарламаның нәтижесі

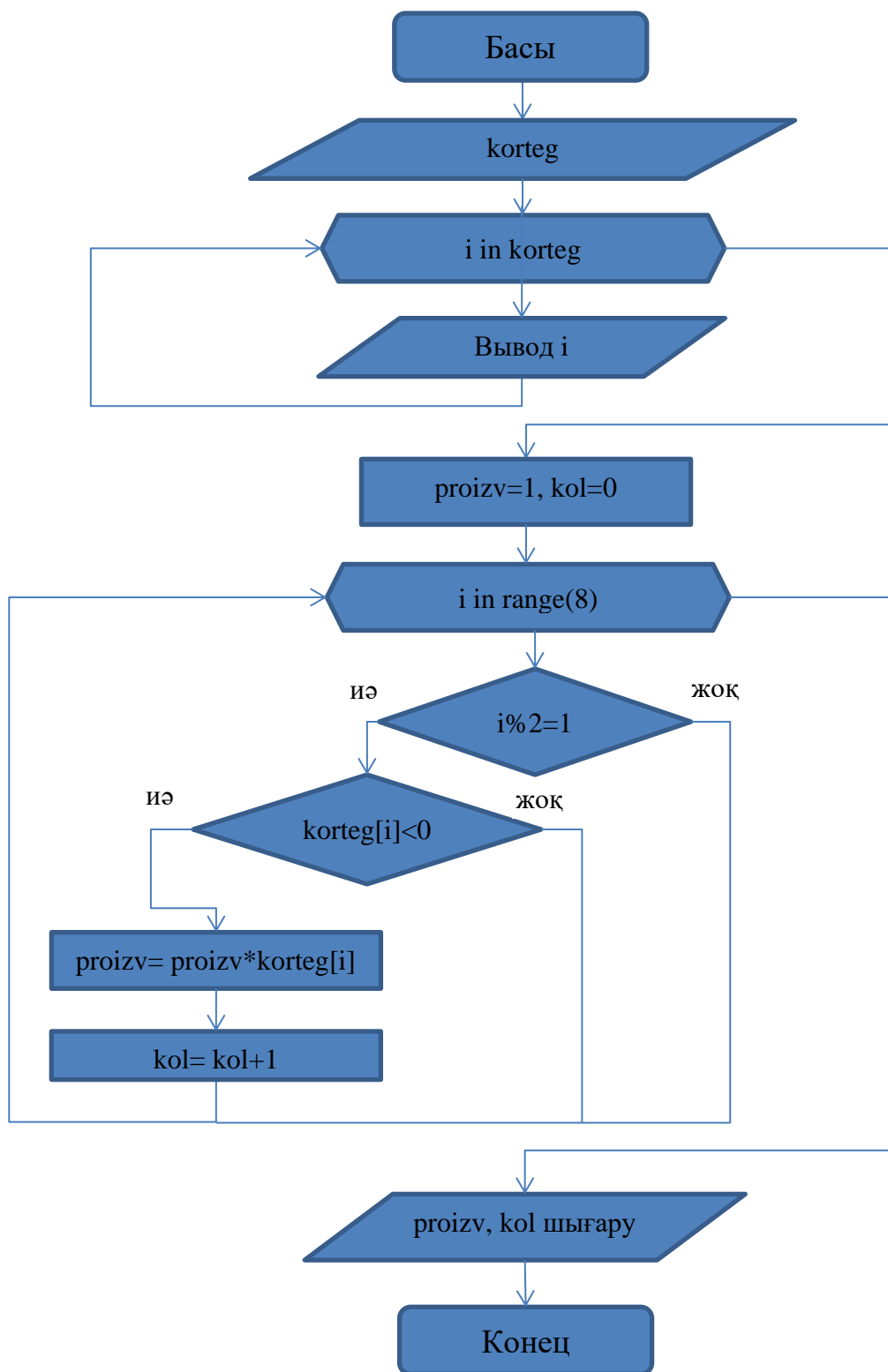
7.5 Есептерді шығаруға арналған мысалдар

Есеп 7.5.1. Бүтін мәнді кортеждің тақ индекстері бар теріс элементтердің көбейтіндісін есептеңіз.

Шешімі. Листингте есепті шешуге жауап беретін бағдарлама коды бар:

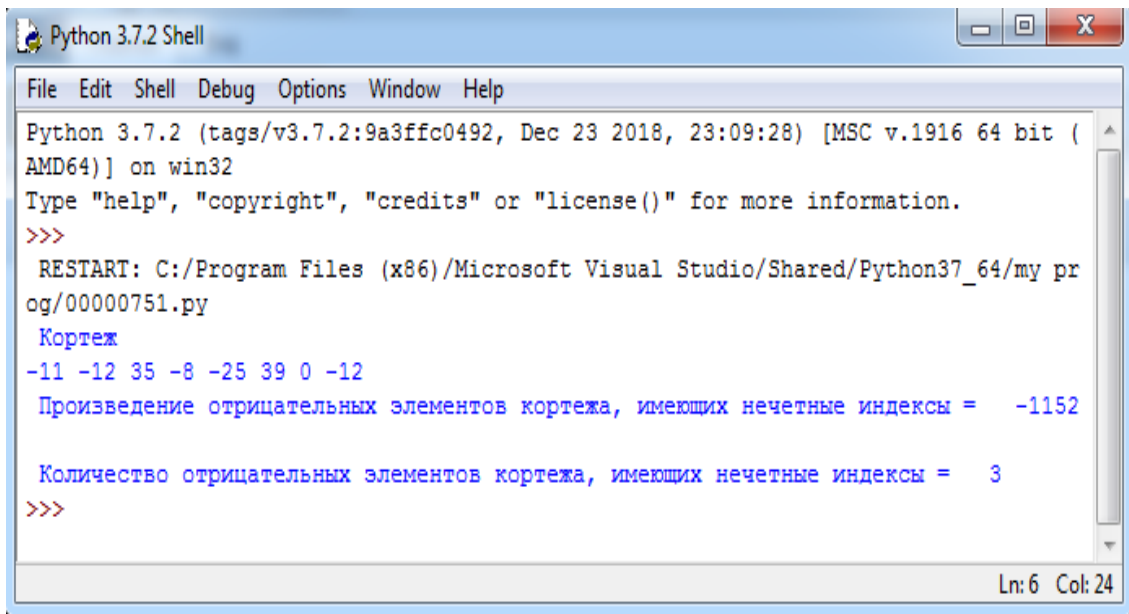
```
korteg=(-11, -12, 35, -8, -25, 39, 0, -12)
print(" Кортеж ")
for i in korteg:
    print(i, end=" ")
kol=0
proizv=1
for i in range(8):
    if i%2==1: # Поиск элементов кортежа с нечетными индексами
        if korteg[i]<0:
            proizvod=proizv*korteg[i] # Произведение отрицательных из них
            kol=kol+1 # Количество отрицательных из них
print("\n Произведение отрицательных элементов кортежа, имеющих
нечетные индексы = ", proizvod)
print("\n Количество отрицательных элементов кортежа, имеющих
нечетные индексы = ", kol)
```

Есепті шешу алгоритмінің блок-схемасы 95-ші суретте көрсетілген.



Сурет 95 – 7.5.1 есепті шешу алгоритмінің блок-схемасы

Нәтиже қорытындысы 96-ші суретте 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 prog/00000751.py
Кортеж
-11 -12 35 -8 -25 39 0 -12
Произведение отрицательных элементов кортежа, имеющих нечетные индексы = -1152
Количество отрицательных элементов кортежа, имеющих нечетные индексы = 3
>>>
Ln: 6 Col: 24
```

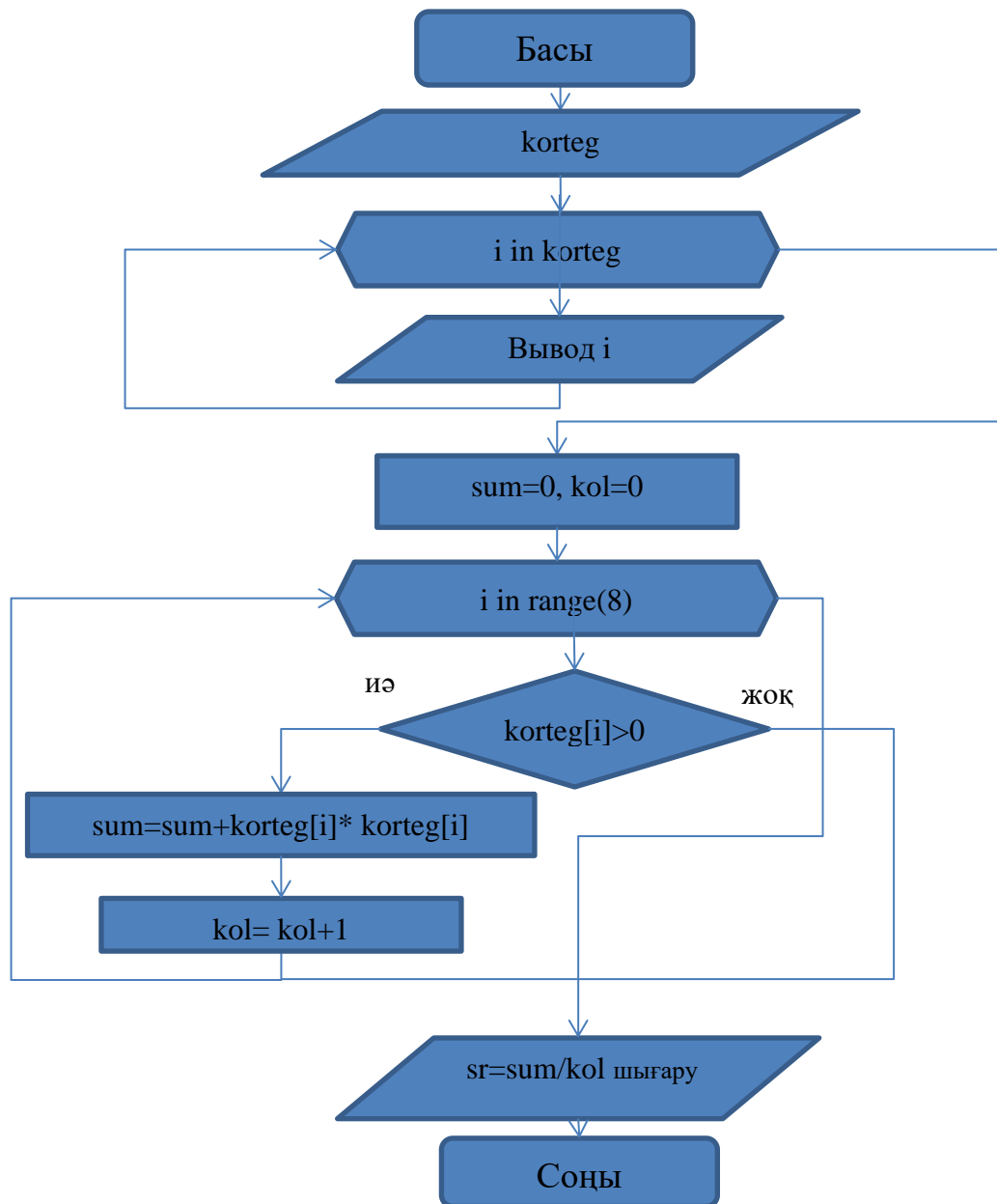
Сурет 96 – 7.5.1 есептің программасының нәтижесі

Есеп 7.5.2. Кортеждегі бүтін сандардың оң элементтердің квадраттарының арифметикалық орташа мәнін есептеңіз.

Шешімі . Листингте есепті шешуге жауап беретін бағдарлама коды бар:

```
korteg=(-11, -12, 35, -8, -25, 39, 0, -12)
print(" Кортеж ")
for i in korteg:
    print(i, end=" ")
kol=0
sum=0
for i in range(8):
    if korteg[i]>0:
        sum=sum+korteg[i]*korteg[i] # Сумма квадратов положительных
элеменов кортежа
        kol=kol+1 # Количество положительных элеменов кортежа
sr=sum/kol
print("\n Среднее арифметическое квадратов положительных элеменов
кортежа = ", sr)
print("\n Количество положительных элеменов кортежа = ", kol)
```

Есепті шешу алгоритмінің блок-схемасы 97-ші суретте көрсетілген.



Сурет 97 – 7.5.2 есептің шешу алгоритмінің блок схемасы

Нәтиже қорытындысы 98-ші суретте 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/00000752.py
Кортеж
-11 -12 35 -8 -25 39 0 -12
Среднее арифметическое квадратов положительных элементов кортежа = 1373.0
Количество положительных элементов кортежа = 2
>>> |
Ln: 10 Col: 4
  
```

Сурет 96 – 7.5.2 есептің қорытынды нәтижесі

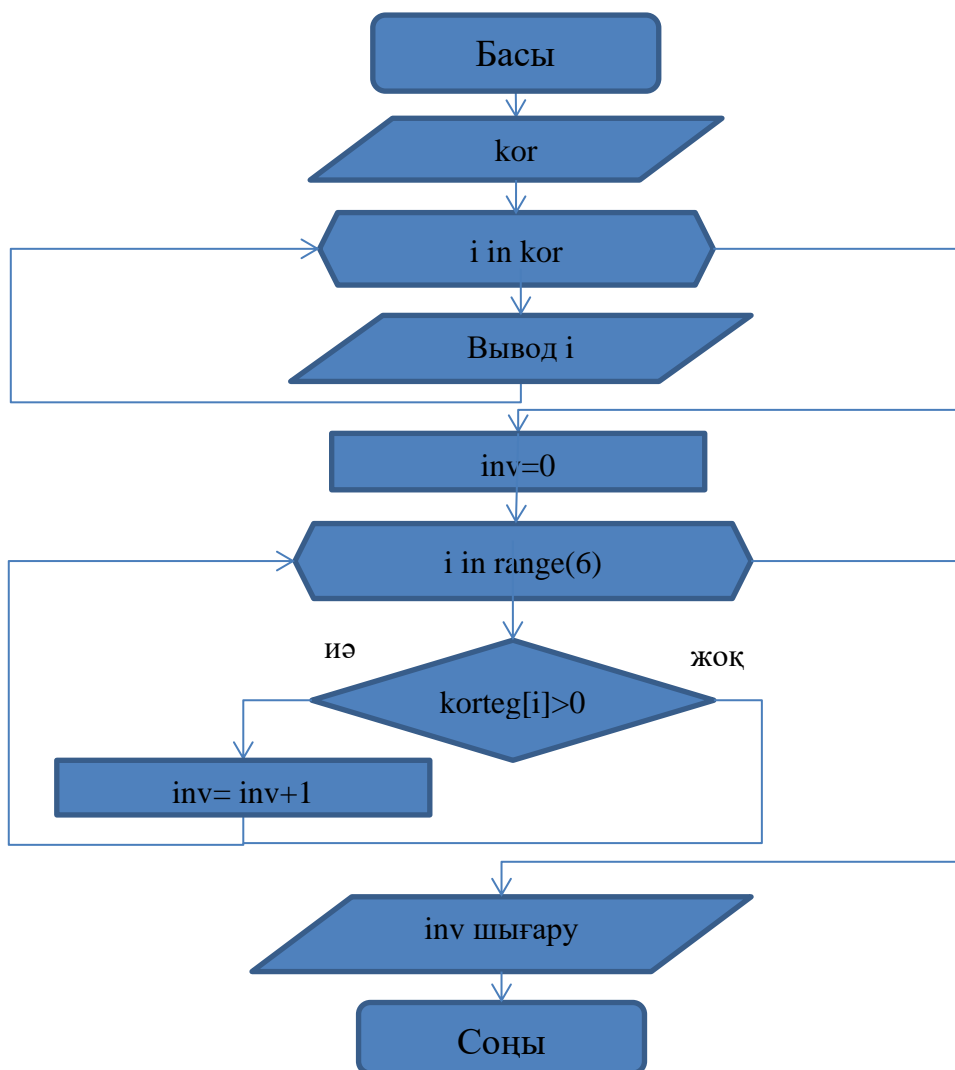
Есеп 7.5.3. Бүтін сандар кортежінде инверсияның санын анықтаңыз. Инверсия дегеніміз - үлкен сан кішіден солға ғарай орналасқан элементтер жұбы. Мысалы, кортеж берілген:



Инверсия саны стрелкамен(бағытауыш) көрсетілген.

Шешімі . 1-ден 6-ға дейінгі циклды ұйымдастырып, тізбектің әр элементін алдыңғысымен салыстырып, оның одан үлкен екенін анықтаймыз. 1-ден 7-ге дейінгі циклды ұйымдастырудың мағынасы жоқ, **өйткені** $kor[i] > kor[i+1]$ тексерілгеннен кейін кортеждің жетінші элементін сегізіншімен салыстырамыз бұл мән бізде жоқ. Егер тексерілген шарт ақиқат болса, онда $inv = inv + 1$ операторының көмегімен есептегішті бірге артырып жауабын шығарамыз.

Есепті шешу алгоритмінің блок-схемасы 97-ші суретте көрсетілген.



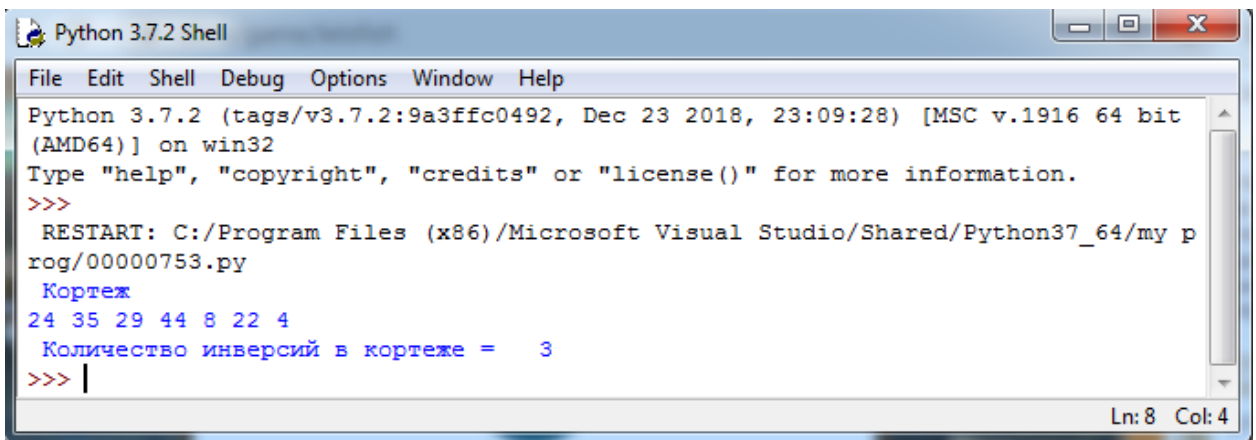
Сурет 97 – 7.5.3 есептің алгоритмінің блок-схемасы

```

Листингте есепті шешуге жауап беретін бағдарлама коды көрсетілген:
kor=(24, 35, 29, 44, 8, 22, 4)
print(" Кортөж ")
for i in kor:
    print(i, end=" ")
inv=0
for i in range(6):
    if kor[i]>kor[i+1]:
        inv=inv+1
print("\n Количество инверсий в кортеже = ", inv)

```

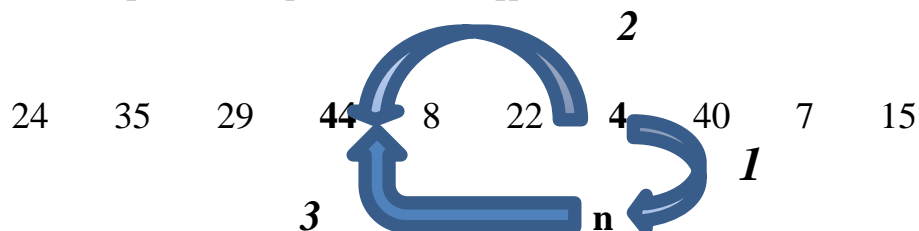
Программаның қорытынды нәтижесі 98-ші суретте көрсетілген



Сурет 98 – 7.5.3 есептің қорытынды нәтижесі

Есеп 7.5.4. Кездейсоқ құрылған он бүтін саннан тұратын тізімде максималды және минималды элементтерді тауып, оларды алмастырыңыз.

Шешімі . Тізім элементтерін ауыстыру үшін бізге **nommax** және **nommin** деген екі ұяшық керек. Бастапқыда біз тізімдегі максималды элементтің саны нөлге тең, ал минималды элементтің саны да нөлге тең деп санаймыз. Осыдан бағдарламада келесі операторлар шығады: **nommax = 0** және **nommin = 0**. Циклдік алгоритмдерді бағдарламалау кезінде максималды және минималды сандарды табу әдістері қарастырылды, сондықтан біз оларға тоқталмаймыз. Ұяшықтарды алмастыру үшін бізге үшінші ұяшық қажет - мысалы, **n**. Бүтін сандардың тізімі құрылды делік:



Тізімдегі минималды элемент - 4, индекс нөмірі - 6, ал тізімдегі максималды элемент - 44, индекс нөмірі 3. n - ұяшығын алайық -. Бірінші

әрекет (1) табылған минималды санды оған көшіру болады. **Екінші әрекет арқылы (2)** максималды санды бос орынға жылжытамыз. Сонымен, **үшінші әрекет арқылы (3)** **n** ұяшығынан минималды сан тізімдегі максимум саны табылған жерге орналастырылады. Бағдарламада бұл әрекеттерді келесі операторлар арқылы көрсетуге болады:

```
n=chislo[6]
chislo[6]=a[3]
a[3]=n
```

Тізімдегі қай элементтің минималды, қайсысының максималды болатындығы және олардың саны қандай болатындығы алдын-ала белгісіз болғандықтан, біз операторлардың мұндай ретімен алмастырамыз:

```
n=chislo[nommin]
chislo[nommin]=chislo[nommax]
chislo[nommax]=n
```

Листингте есепті шешуге жауап беретін бағдарлама коды көрсетілген:

```
import random
chislo=random.sample(range(100), 10)
print(chislo)
nommin=0 # Считаем, что первоначально номер минимального элемента
в списке 0
nommax=0 # Считаем, что первоначально номер максимального
элемента в списке 0
min=32767
max=-32768
for i in range(1, 10):
    if chislo[i]<min: # Поиск минимального элемента в списке
        min=chislo[i]
        nommin=i # Поиск индекса минимального элемента в списке
    if chislo[i]>max: # Поиск максимального элемента в списке
        max=chislo[i]
        nommax=i # Поиск индекса максимального элемента в списке
print("Минимальное число = ", min)
print("Максимальное число = ", max)
n=chislo[nommin] #
chislo[nommin]=chislo[nommax]
chislo[nommax]=n
for i in range(0, 10):
    print(chislo[i], end='\t')
```

Бағдарламаның нәтижесі 99-ші суретте көрсетілген

```

PS C:\Windows.old\Users\over1\Desktop\Python> & C:/Users/user/AppData/Local/Programs/Python/Python39/python.exe c:/Windows.old/Users/over1/Desktop/Python/importrandom.py
[5, 85, 52, 77, 20, 83, 44, 35, 82, 65]
Минимальное число = 20
Максимальное число = 85
5 20 52 77 85 83 44 35 82 65

```

Сурет 99 – 7.5.4 есептің қорытынды нәтижесі

Есеп 7.5.5. Нақты сандардан тұратын кортеж берілген. Кортеждің оң элементтерін тізімге жазыңыз. Оң элементтердің санын анықтаңыз. Тізім элементтерінің көбейтіндісін табыңыз.

Шешімі. Бастапқыда жасалған кортежде жеті нақты сан бар, оның төртеуі оң. Бос тізім `spisok = []` операторының көмегімен жарияланады. Егер оператор үлкен болса `0 < kort [i] < 0` кортежге енгізілген санның оңдылығын тексереді, ал егер ол ақиқат болса, оны `append ()` әдісі арқылы тізімге қосамыз. `N` оң сандар санының есептегішін `n = n + 1` операторы бірге артырады .

Содан кейін, `n` ұяшығында сақталған оң сандардың белгілі саны бар цикл ұйымдастырылады және оның ішінен біз тізім элементтерінің көбейтіндісін табамыз, оларды экранға шығарамыз.

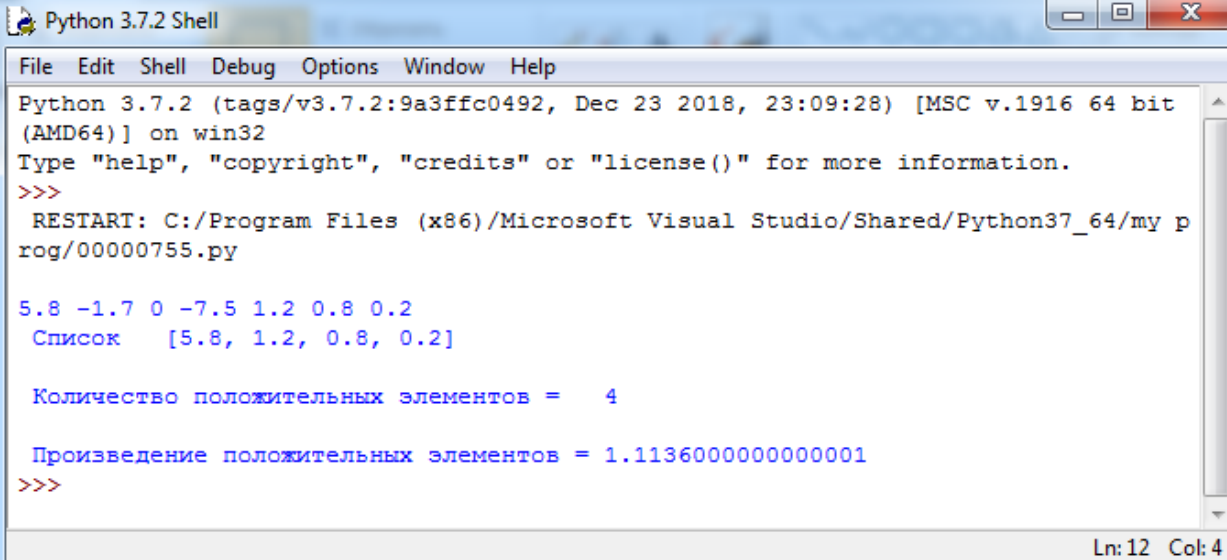
Төменде есепті шешуге жауапты бағдарлама коды көрсетілген.

```

kort=(5.8, -1.7, 0, -7.5, 1.2, 0.8, 0.2)
spisok=[]
n=0
proizv=1
print("")
for i in kort:
    print(i, end=" ")
for i in range(7):
    if kort[i]>0:
        spisok.append(kort[i])
        n=n+1
for i in range(n):
    proizvod=proizv*spisok[i]
print("\n Список ", spisok)
print("\n Количество положительных элементов = ", n)
print("\n Произведение положительных элементов =", proizvod)

```

Нәтижесі 100-ші суретте 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/00000755.py

5.8 -1.7 0 -7.5 1.2 0.8 0.2
Список [5.8, 1.2, 0.8, 0.2]

Количество положительных элементов = 4

Произведение положительных элементов = 1.1136000000000001
>>>
Ln:12 Col:4
```

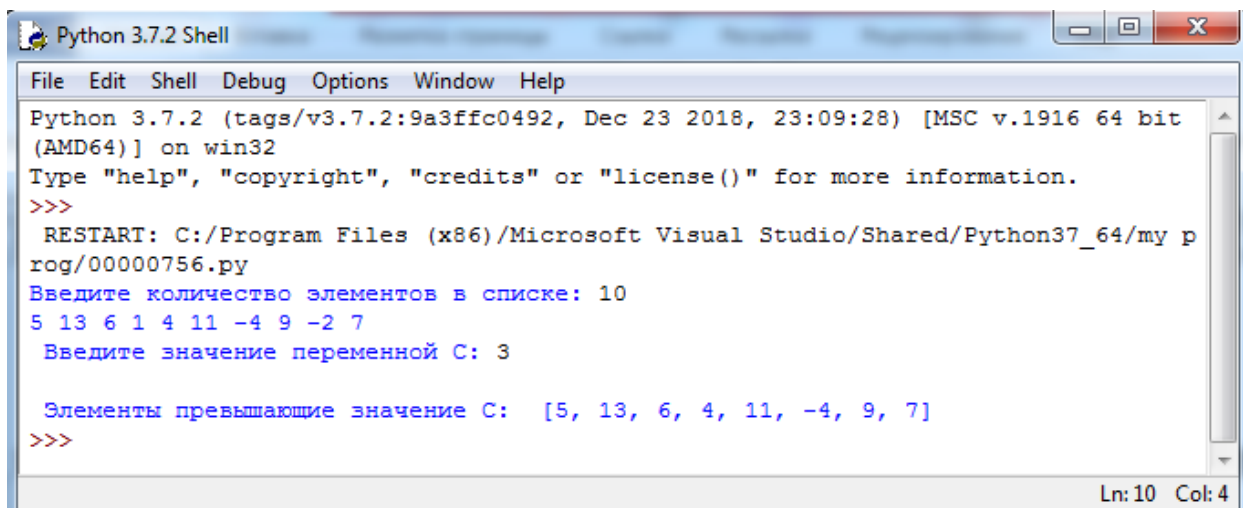
Сурет 100 – 7.5.5 есептің нәтижесі

Есеп 7.5.6. N элементтер тізімін әртүрлі белгілердің кездейсоқ бүтін сандарымен толтыратын бағдарлама құрыңыз. Құрылған тізімді компьютер экранына шығарыңыз және модульдегі элементтері енгізілген C мәнінен үлкен болған элементтерден басқа тізімді құрыңыз.

Шешімі . Бастапқыда біз пайдаланушы көрсеткен элементтер санын қайтаратын **sample** функциясын қолдана отырып сандар тізбегін құрамыз. Содан кейін, C мәнін енгізгеннен кейін, y тізімін инициализациялаңыз, **abs(x[i])>C** логикалық өрнегін тексеретін циклды ашыңыз және егер ол ақиқат болса, **append ()** әдісін қолдана отырып, y тізіміне бастапқы реттік элементін қосыңыз.

Листингте есепті шешуге жауап беретін бағдарлама коды бар.

```
import random
from math import *
n=int(input("Введите количество элементов в списке: "))
x=random.sample(range(-5, 15), n)
for i in x:
    print(i, end=" ")
c=int(input("\n Введите значение переменной C: "))
y=[]
for i in range(n):
    if abs(x[i])>c:
        y.append(x[i])
print("\n Элементы превышающие значение C: ", y)
Бағдарламаның нәтижесі 101-ші суретте 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/00000756.py
Введите количество элементов в списке: 10
5 13 6 1 4 11 -4 9 -2 7
Введите значение переменной C: 3
Элементы превышающие значение C: [5, 13, 6, 4, 11, -4, 9, 7]
>>>
Ln: 10 Col: 4
```

Сурет 101 – 7.5.6 есебінің қорытынды нәтижесі

Есеп 7.5.7. N студенттерінің есімдерінің бастапқы тізімін жасаңыз. Алынған екі тізімді бастапқы тізімнің жұп және тақ нөмірлері бойынша қалыптастыруды ұйымдастырыңыз. Тақ сандары бар тізімді кему бойынша, жұп сандары - өсу бойынша реттеңіз

Шешімі . Біз үш бос тізімді жасаймыз (**spisok**, **chetn**, **nchetn**) және студенттердің аттарын енгізуді, сондай-ақ оларды бастапқы тізімге **append()** әдісімен қосуды ұйымдастырамыз. Келесі циклде % (модуль бойынша бөлу) операциясын пайдалана отырып, Элемент нөмірін паритетке тексеруді жүзеге асырамыз. Егер қалдық нөлге тең болса, онда элемент жұп сандар тізіміне қосылады, әйтпесе тақ сандар тізімі жасалады. Тізімдерді ретке келтіру әдістерін қолдана отырып (б-кестеге сәйкес), біз оларды экранға шығарамыз.

Листингте есепті шешуге жауап беретін бағдарлама коды бар.

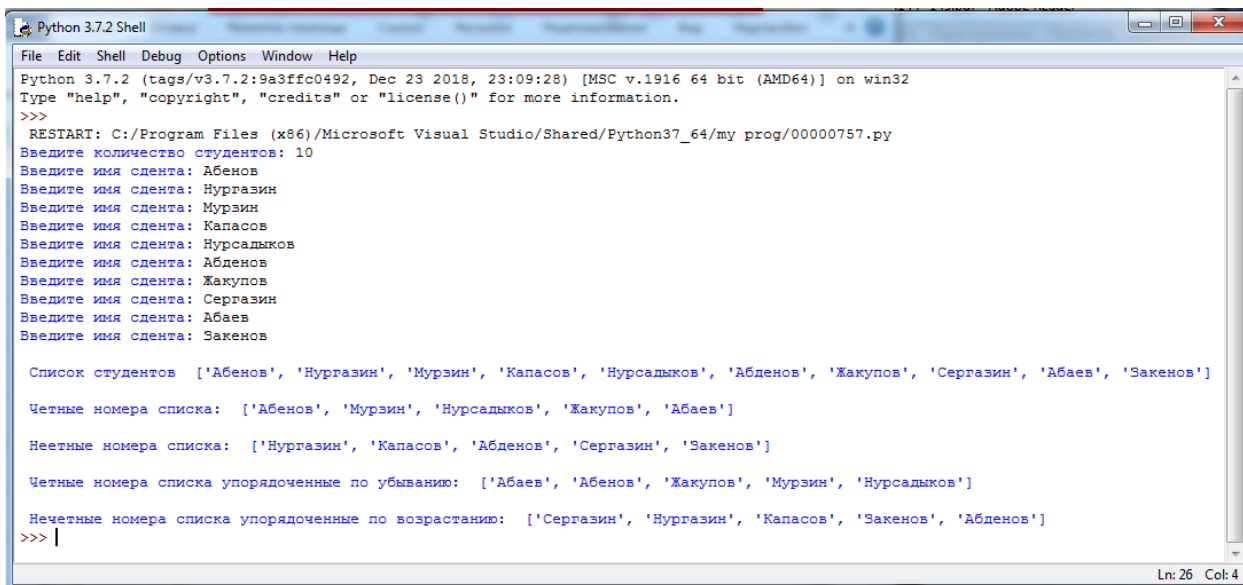
```
n=int(input("Введите количество студентов: "))
spisok=[]
chetn=[]
nchetn=[]
for i in range(n):
    sp=input("Введите имя студента: ")
    spisok.append(sp)
print("\n Список студентов ", spisok)
for i in range(n):
    if i%2==0:
        ch=spisok[i]
        chetn.append(ch)
    else:
        nech=spisok[i]
        nchetn.append(nech)
print("\n Четные номера списка: ", chetn)
```

```

print("\n Нечетные номера списка: ", nechtn)
for i in chetn:
    chetn.sort()
print("\n Четные номера списка упорядоченные по убыванию: ", chetn)
for i in nechtn:
    nechtn.sort(reverse=True)
print("\n Нечетные номера списка упорядоченные по возрастанию: ",
nechetn)

```

Бағдарламаның нәтижесі 102-ші суретте көрсетілген.



Сурет 102 – 7.5.7 есебінің қорытынды нәтижесі

7.6 Бақылау сұрақтары

- 1.Кортеждердің басты ерекшелігі неде?
- 2.Кортежді бағдарламаларда пайдаланудың қандай артықшылықтар бар?
3. Кортежді жариялаудың синтаксисін жазыңыз.
4. Кортежді өңдеу кезінде оның әрбір элементіне қол жеткізу қалай жүзеге асырылады?
5. Кортеждерді өңдеудің классикалық әдістерін атаңыз.
6. Бағдарламада кортежді қалай біріктіруге(срез) болады?
- 7.Кортеж элементтерінің мәндерінің алмасуы қалай жүзеге асырылатындығын түсіндіріңіз.
8. Python тілінде жасалған тізімдер мен кортеждердің арасындағы айырмашылық неде екенін түсіндіріңіз.
9. Тізімді жариялау синтаксисін жазыңыз.
10. Тізімдерді құру үшін Python тілінің қандай мүмкіндіктері қолданылады?
11. Тізімдермен жұмыс істеудің негізгі әдістерін атаңыз және түсіндірме

беріңіз.

12. Python тілінің деректер құрылымына анықтама беріңіз, сөздік сияқты.

13. Сөздік құрудың синтаксисін жазыңыз.

14. Сөздікті құру кезінде қандай ережелерді қолдану керек?

7.7 Өздігінен шешуге арналған есептер

1. В кортеже целых чисел вычислите произведение отрицательных элементов, имеющих нечетные индексы.

2. Из исходного списка целых чисел сформируйте два списка: список четных чисел **V** и список нечетных чисел **C**.

3. Определите среднее арифметическое элементов кортежа, удовлетворяющих условию **abs(korteg[i])>C**. Значение **C** вводится с клавиатуры.

4. Разработайте программу, в которой определяются максимальный и минимальный элементы среди положительных нечетных элементов целочисленного кортежа **A(10)**.

5. Разработайте программу, заполняющую список из **N** элементов случайными целыми числами, находящимися в интервале от **-20** до **40**. Выведите на экран компьютера созданный список. В списке положительные элементы уменьшите вдвое, а отрицательные замените на значения их индексов.

6. Разработайте программу, заполняющую список из **N** элементов случайными целыми числами, находящимися в интервале от **1** до **40**. Определите, сколько процентов всего количества элементов списка составляют нечетные элементы.

7. Разработайте программу, заполняющую список из **N** элементов случайными целыми числами, находящимися в интервале от **1** до **50**. Выведите на экран компьютера созданный список и упорядочите элементы данного списка по возрастанию их значений.

8. Из списка произвольных чисел **A[10]** сформируйте другой список таким образом, чтобы вначале были отрицательные элементы исходного списка, затем положительные и, наконец, нулевые.

9. Разработайте программу, заполняющую список из **N** элементов случайными целыми числами, находящимися в интервале от **1** до **30**. Выведите на экран компьютера созданный список и найдите максимальный элемент, его номер и поменяйте местами максимальный и первый элемент списка.

10. Разработайте программу, которая включает заданное число **D** в список **A[10]**, упорядоченный по возрастанию, с сохранением упорядоченности.

11. Разработайте программу, в которой удалите из списка **A**, состоящего из **n** элементов, первые четыре нулевых элемента.

12. Разработайте программу, заполняющую список из N элементов случайными целыми числами, находящимися в интервале от -30 до 50 . Выведите на экран компьютера созданный список и определите, есть ли в нем серии элементов, состоящих из знакопередающихся чисел. Если есть, то выведите на экран количество таких серий.

13. Разработайте программу, которая выводит на экран два кортежа $A(10)$, содержащих диаметры и веса шин. Следует отобрать две шины, диаметры которых отличаются не более чем на D см, а вес - не более чем на W килограмм.

14. Из списка произвольных чисел $A[10]$ сформируйте два списка, содержащих номера положительных и отрицательных элементов.

15. Разработайте программу, заполняющую список из N элементов случайными целыми числами, находящимися в интервале от -20 до 40 . Выведите на экран компьютера созданный список и вычислите среднее арифметическое значение квадратов положительных элементов.

16. В кортеже целых чисел произведите обмен соседних элементов, стоящих на четных местах, с элементами, стоящими на нечетных местах.

17. Разработайте программу, заполняющую список из N элементов случайными вещественными числами, находящимися в интервале от 1 до 30 . Все элементы списка с четными номерами, предшествующие первому по порядку элементу с наибольшим значением, домножьте на него.

18. Разработайте программу, заполняющую список из N элементов случайными целыми числами, находящимися в интервале от -15 до 20 . Выведите на экран компьютера созданный список и найдите наибольший элемент из отрицательных.

19. Разработайте программу, заполняющую список из N элементов случайными целыми числами, находящимися в интервале от 1 до 50 . Выведите на экран компьютера созданный список и найдите количество тех элементов, значения которых находятся в диапазоне от A до B .

20. Пользователь вводит с клавиатуры элементы списка $A[n]$. Определите, является ли заданная последовательность чисел a_1, a_2, \dots, a_N монотонно убывающей.

21. Разработайте программу, заполняющую список из N элементов случайными целыми числами, находящимися в интервале от 1 до 30 . Замените нулями элементы между максимальным и минимальным значениями, кроме них самих.

22. Разработайте программу, в которой в кортеже целых чисел требуется найти индекс последнего по счету отрицательного элемента.

23. Разработайте программу, которая определяет, имеется ли в заданном целочисленном кортеже $A(10)$ хотя бы одна пара совпадающих по значению чисел.

24. Разработайте программу, которая выводит на экран два кортежа, содержащих кортежи ростов игроков двух команд (в см), и определяет, имеется ли в данных командах игроки одинакового роста.

25. Разработайте программу, заполняющую список из N элементов случайными целыми числами, находящимися в интервале от **1** до **50**. Выведите на экран компьютера созданный список и найдите максимальный и минимальный элементы, вычислите их разность.