

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" жұбындағы 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.

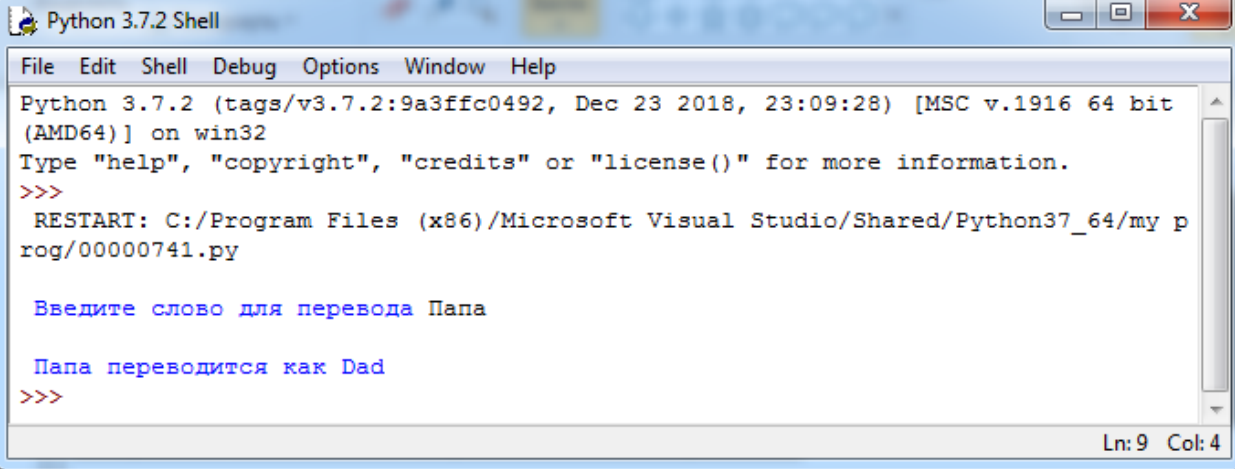
Сөздік көмегімен бағдарламалардың бірнеше мысалын қарастырыңыз.

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
>>>
```

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",

```



```

        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
>>>
Ln: 14 Col: 84

```

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:
            proizv=proizv*korteg[i] # Произведение отрицательных из них

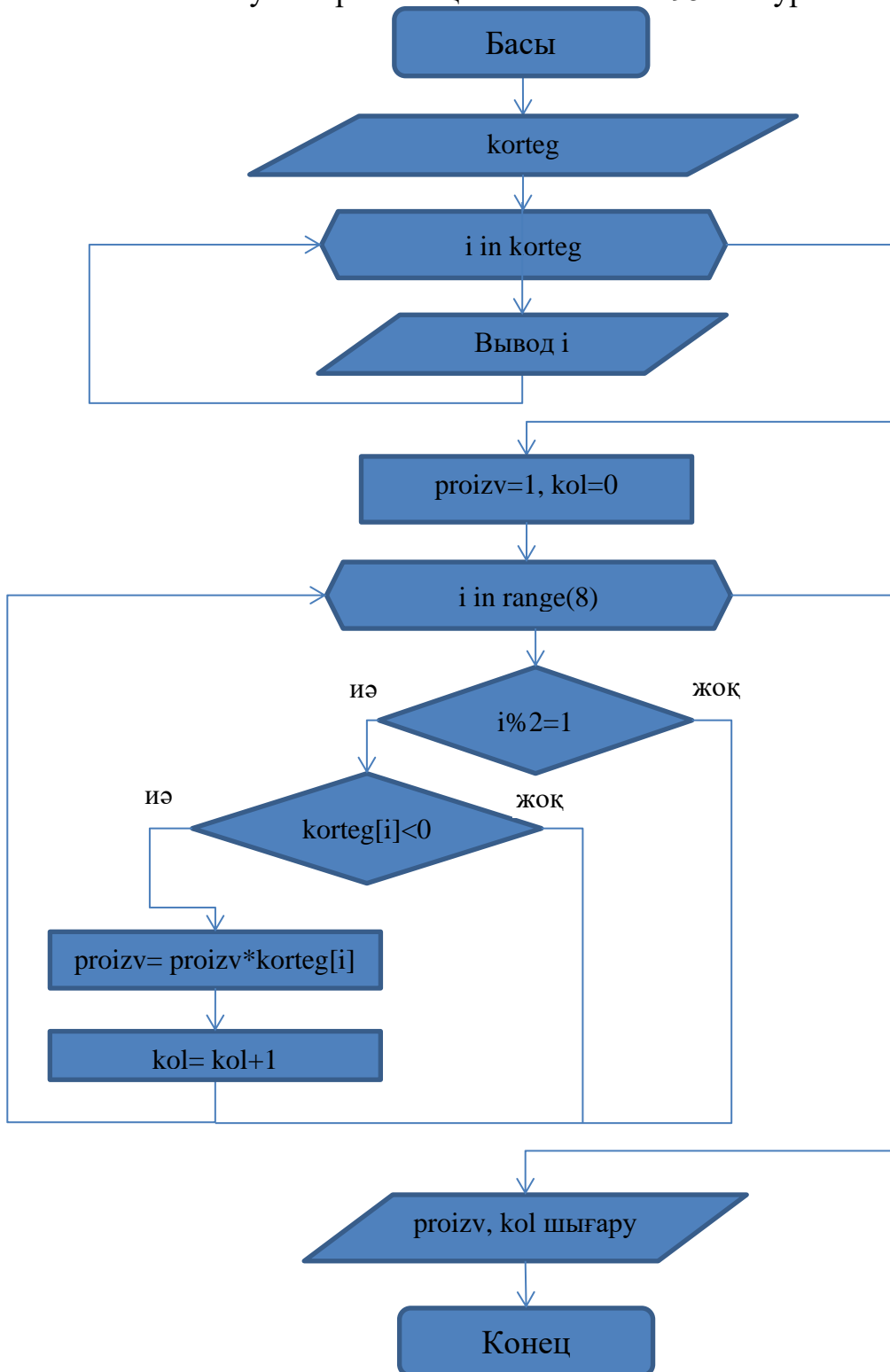
```

```

kol=kol+1 # Количество отрицательных из них
print("\n Произведение отрицательных элементов кортежа, имеющих
нечетные индексы = ", proizv)
print("\n Количество отрицательных элементов кортежа, имеющих
нечетные индексы = ", kol)

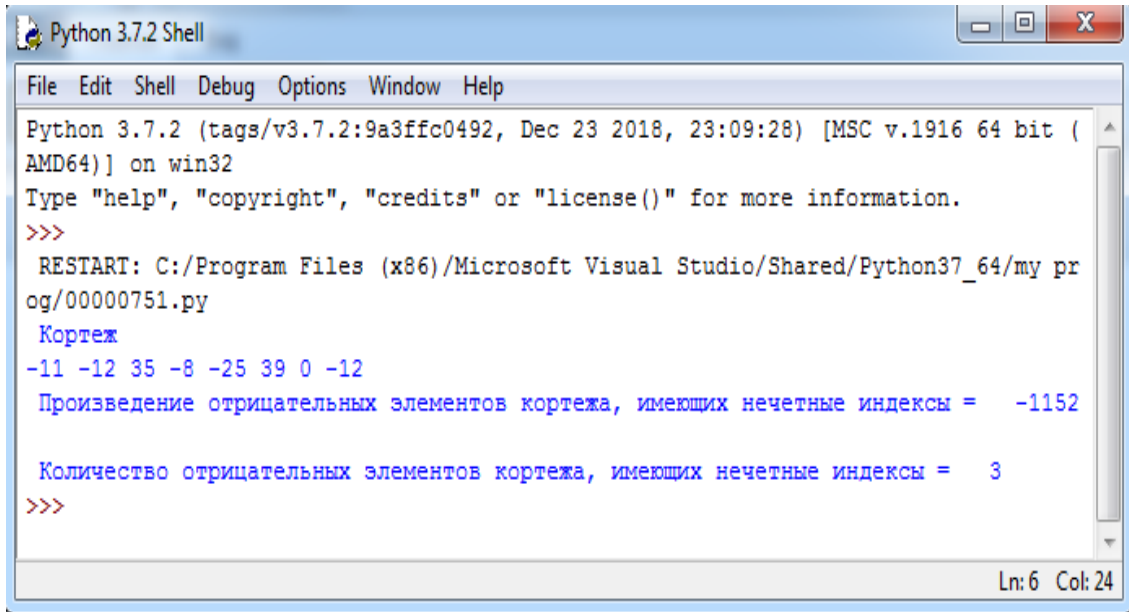
```

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
```

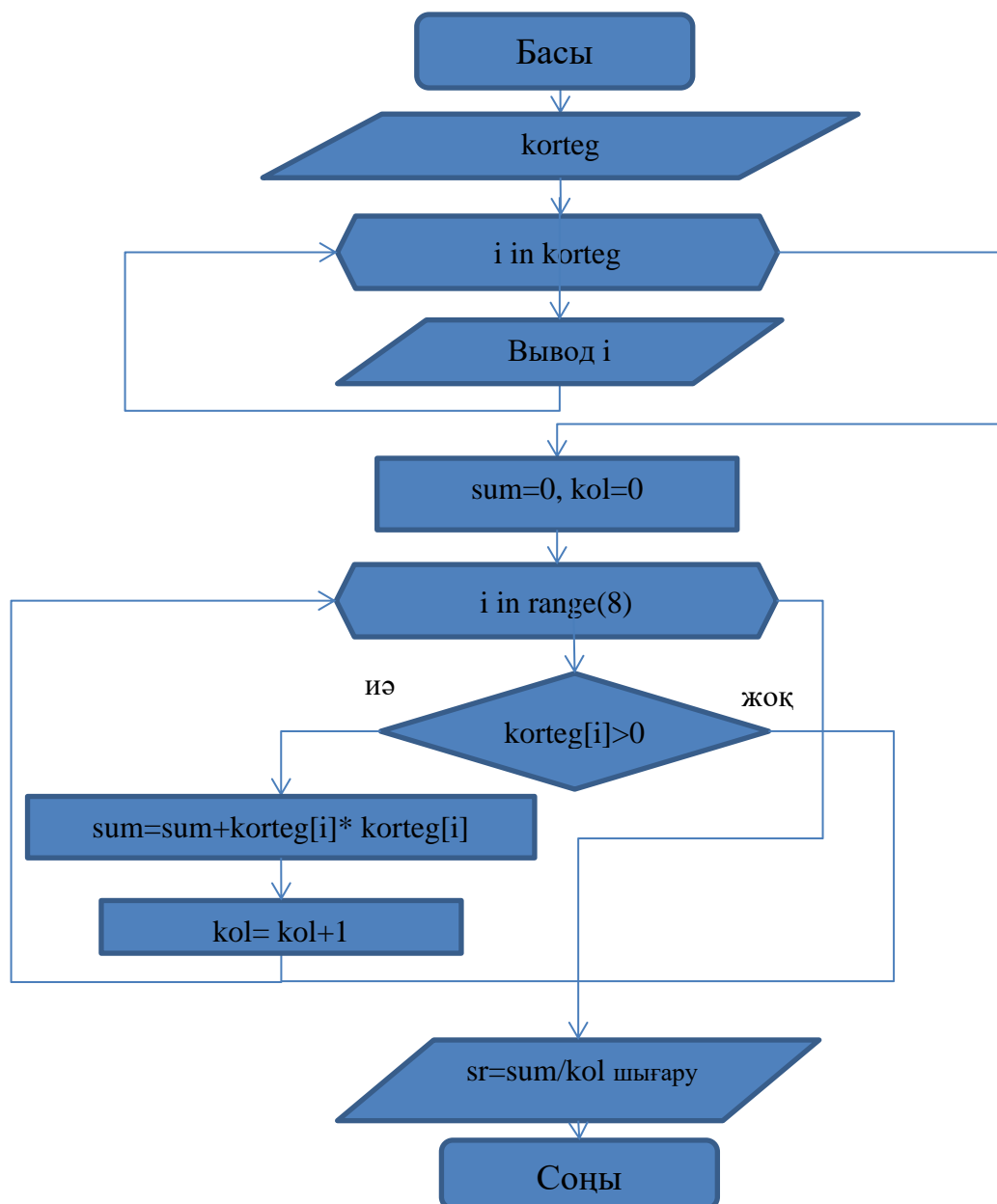
Сурет 96 – 7.5.1 есептің программасының нәтижесі

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

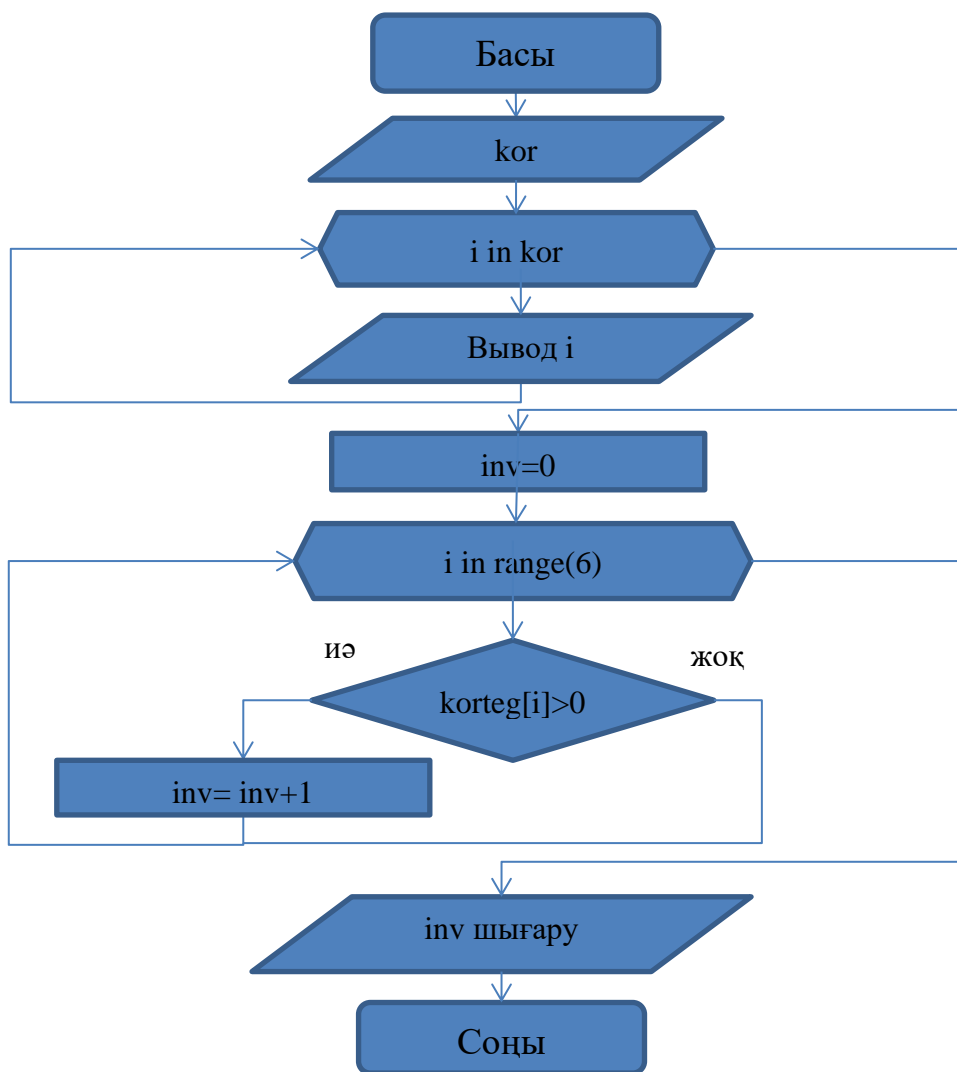
Есеп 7.5.3. Бүтін сандар кортежінде инверсияның санын анықтаңыз. Инверсия дегеніміз - үлкен сан кішіден солға ғарай орналасқан элементтер жұбы. Мысалы, кортеж берілген:



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

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.

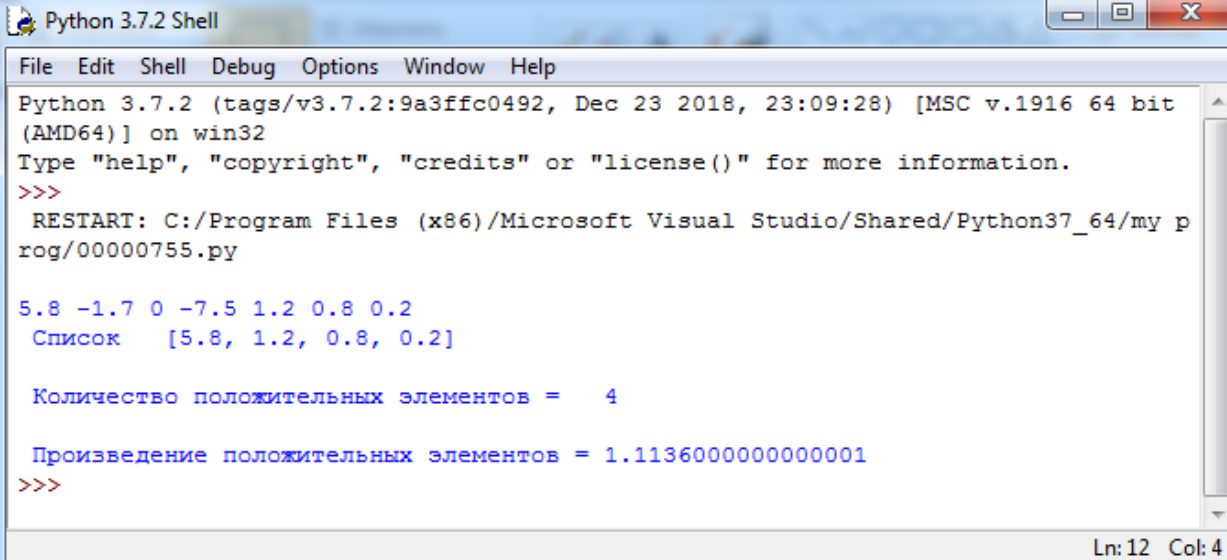
Төменде есепті шешуге жауапты бағдарлама коды көрсетілген.

```

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
```

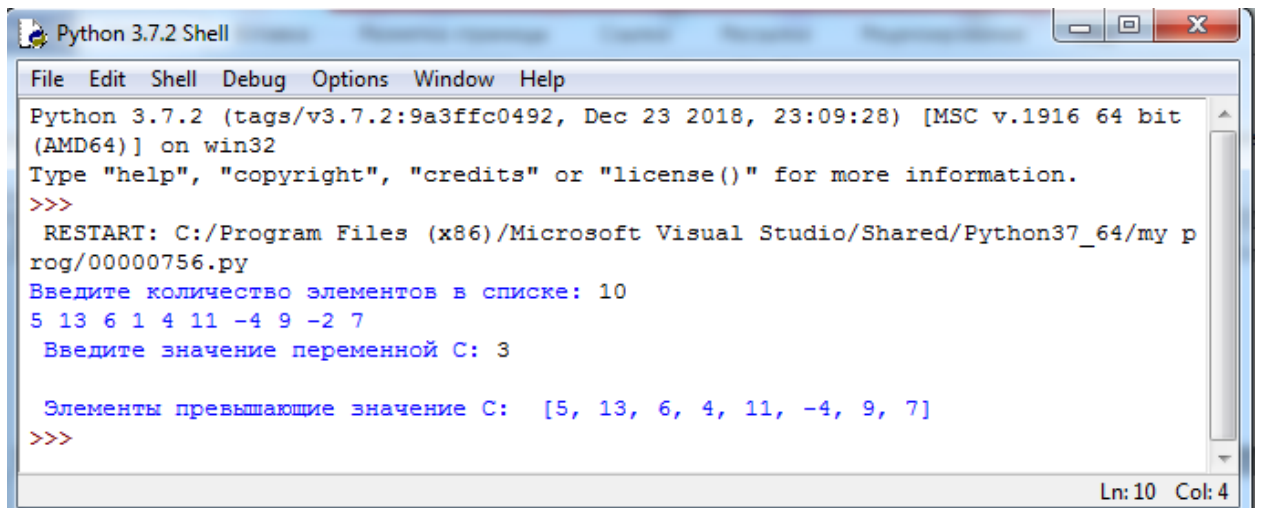
Сурет 100 – 7.5.5 есептің нәтижесі

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 реттілік 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.

```

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/00000757.py
Введите количество студентов: 10
Введите имя студента: Абенов
Введите имя студента: Нургазин
Введите имя студента: Мурзин
Введите имя студента: Капасов
Введите имя студента: Нурсаджков
Введите имя студента: Абденов
Введите имя студента: Жакупов
Введите имя студента: Сергазин
Введите имя студента: Абаев
Введите имя студента: Закинов

Список студентов ['Абенов', 'Нургазин', 'Мурзин', 'Капасов', 'Нурсаджков', 'Абденов', 'Жакупов', 'Сергазин', 'Абаев', 'Закинов']

Четные номера списка: ['Абенов', 'Мурзин', 'Нурсаджков', 'Жакупов', 'Абаев']

Нечетные номера списка: ['Нургазин', 'Капасов', 'Абденов', 'Сергазин', 'Закинов']

Четные номера списка упорядоченные по убыванию: ['Абаев', 'Абенов', 'Жакупов', 'Мурзин', 'Нурсаджков']

Нечетные номера списка упорядоченные по возрастанию: ['Сергазин', 'Нургазин', 'Капасов', 'Закинов', 'Абденов']
>>> |
Ln: 26 Col: 4

```

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.