

ШЫҒЫС ҚАЗАҚСТАН ГУМАНИТАРЛЫҚ КОЛЛЕДЖІ



PYTHON ПРОГРАММАЛАУ ТІЛІНДЕ АЛГОРИТМДЕРДІ ПРОГРАММАЛАУ

Касенова А.А.

Өскемен қаласы, 2019ж

ӘӨЖ 004(075 32)
ҚБЖ 73ж75
К21

Шығыс Қазақстан Гуманитарлық колледжінің ғылыми әдістемелік кеңесі
отырысында қарастырылды және мақұлданды.

Хаттама №____ 2019ж.

Ғылыми әдістемелік кеңесінің төрайымы Ж.Слямбеков.

Пікір жазған: Аршын Ж. -ШҚГК информатика және арнайы пәндер
оқытушысы

Python программалау тіліне әзірленген практикалық тапсырмалар жинағы /
А.А. Касенова, Өскемен қаласы, 2019 ж. 26 Бет

Ұсынылып отырған әдістемелік құрал Python программалау тілін жетік
білуге арналған.

© Касенова А.А. 2019ж.
© ШҚГК 2019ж.

Мазмұны

Алғысөз.....	5
Python программалау тілі.....	6
Массив.....	8
Қорытынды сұрақтар.....	17
Python программалау тілінде практикалық тапсырмалар.....	18
Тест.....	24
Глоссарий.....	26
Қорытынды.....	26
Пайдаланылған әдебиеттер, электрондық қорлар.....	26

АЛҒЫСӨЗ

«Python программалау тілі» практикалық және зертханалық жұмыстарды орындауға арналған әдістемелік нұсқаулық. Python программалау тілінде өз білімдеріңді кеңейте түсесіңдер. Python-бір мезгілде қарапайым және қуатты объектілі-бағытталған бағдарламалау тілі болып табылады. Python жоғары сыныпта қолданылатын программалау тілі. ХХІ ғасырдан келе жатқан параллель есептеулер күрделі және қол жетімсіз болатын. Python тілі мен қоса көптеген платформалар мен тілдер параллелизацияны жүзеге асыруға мүмкіндік береді. Python көптеген тапсырмаларды орындауға арналған көптеген құралдарды ұсынады: көп дәрежелі, көп процедуралы. Қысқаша айтқанда, математикалық есептеулер үшін көп функциялы мультиқауіпсіздікті қолдануға болады



Python программалау тілі

"Python"— ең жаңа, заманауи бағдарламалау тіліне жатады. Бұл тілді ең алғашында Ван Россум атты нидерландық адам ойлап тапқан. Python — жоғары дәрежедегі бағдарламалау тілі. Python тілі қазіргі кезде ең көп таралған бағдарламалау тілі болып саналады. Бұл тілде жасалған ең атақты бағдарлама ол бәріміз білетін Instagram әлеуметтік желісі. Бұдан басқа Python бағдарламалау тілінде BitTorrent, Ubuntu, Blender, GIMP, Civilization IV, Battlefield 2, World of Tanks секілді бағдарламаларда кездестіре аламыз. Python-ды қолданатын компанияларға Google, Facebook, Yahoo, NASA, Red Hat, IBM, Instagram, Dropbox, Pinterest, Quora, Яндекс, Mail.Ru т.б келтіре аламыз.

Ең алғаш рет Ван Руссом Python тілін жасауды 1898-жылдың желтоқсан айында қолға алды. Алғашында Python тілі Ameoba операциялық жүйесімен әрекет етуге қабілетті ABC бағдарламалау тілінің ұрпағы ретінде алынған. Кейінірек Python дамып, ең басты және маңызды бағдарламалау тіліне айналды. Ең алғаш жаңарту Python 2.0 жаңа нұсқасы жарық көрді. Біл жаңа нұсқада көптеген функциялар мен жаңа алгоритмдер түрін көрсетті. Жаңартудан кейін Python тілінің қолданушылары арта түседі. Алайда бұл тілдің жаңарту бұнымен біткен жоқ. 2008 жылдың 3 желтоқсанында Python 3.0 жарық көреді.

Python-ның басқа бағдарламалау тілінен ерекшелігі.

Жоғарыда айтып өткендей Python тілі ABC тіліне ұқсас болып келеді. Қазіргі таңда көптеген бағдарламалау тілдерінде бірнеше алгоритмді бір функцияның ішіне салу үшін жақша({}) қойылады. Ал Python тілінде функциядан кейін қос нүкте қойылып, алгоритмдер келесі қатардан бірнеше бос орын қалдырылып жазылады. Сонымен қатар басқа бағдарламалау тілдерінен басты ерекшелігі Python тілінде ақпарат түрлерінің(типы данных) үлкен орын алады. Сол себепті int , double секілді ақпарат түрлеріне үлкен мәнді сандар сия алады.

Python -ның түрлері мен деректер құрылымы

Python-ның айнымалы түрі тек бағдарламаны орындау кезінде анықталады, динамикалық теруді қолдайды. Сондықтан оның орнына «айнымалы тағайындау» шамамен айтуға жақсы «кейбір атымен міндетті күші құндылықтар». Python-ның кірістірілген түрлері: логикалық, жолды, Unicode-жолды, бүтін, еркін-дәлдігін, қалқымалы нүктелі нөмірін, комплекс санды, және басқалар. Python-ның жаңа түрін қосу, сіз класс (класс) жазыңыз, немесе (мысалы, C -жазылған) кеңейту модулінің жаңа түрін анықтауға болады. Сыныптар мұралық (бір және бірнеше) және метобағдарламалық әрекетті қолдайды. Ең кіріктірілген түрлері мұралық болып есептеледі. Барлық нысандар сілтеме және атом болып бөлінеді. Атом бойынша INT және long күрделі және кейбір басқалар болады. Атом заттарды тағайындау кезінде ғана

объектіге анықтамалық көрсеткіш үшін көшіріледі, ал олардың мәні көшіріледі, сондықтан бірдей мәнді тағайындау кезінде екі айнымалы қолданылады. Анықтамалық нысандар өтпелі және даусыз болып табылады. Айнымалы - Мысалы, жолдар және луын өзгермейтін және тізімдер, сөздіктер және басқа да көптеген нысандар болып табылады. Python жылы Tuple, шын мәнінде, өзгеріссіз тізімі болып табылады. Көптеген жағдайларда, луын жылдам тізімдері сондықтан сіз ретін өзгертуді жоспарлап болмаса, оларды пайдалану үшін үздік болып табылады.

Python -ның мүмкіндіктері

Lisp және Прологта сияқты, Python интерпретаторы операторлар пернетақтадан енгізіледі, онда интерактивті режим, дереу орындалады бар, және нәтижесі (REPL) көрсетіледі. Бұл режим бастаушы ғана емес, қызықты, бірақ сондай-ақ, сіз басты бағдарлама оны пайдаланар алдында интерактивті кез келген аймақ кодын тексеру, немесе жай ғана функцияларын үлкен жиынтығы бар калькулятор ретінде пайдалануға болады деп есептейді тәжірибелі бағдарламашылар.

Python-ның жаңартулары

- Python 1.0 - қантар, 1994 жыл
- Python 1.5 - 31-ші желтоқсан, 1997 жыл
- Python 1.6 - 5-ші қыркүйек, 2000 жыл
- Python 2.0 - 16-шы қазан, 2000 жыл
- Python 2.1 - 17-ші сәуір, 2001 жыл
- Python 2.2 - 21-ші желтоқсан, 2001 жыл
- Python 2.3 - 29-шы шілде, 2003 жыл
- Python 2.4 - 30-шы қараша, 2004 жыл
- Python 2.5 - 19-шы қыркүйек, 2006 жыл
- Python 2.6 - 1-ші қазан, 2008 жыл
- Python 2.7 - 3-ші шілде, 2010 жыл
- Python 3.0 - 3-ші желтоқсан, 2008 жыл
- Python 3.1 - 27-ші шілде, 2009 жыл
- Python 3.2 - 20-шы ақпан, 2011 жыл
- Python 3.3 - 29-қыркүйек, 2012 жыл
- Python 3.4 - 16-шы наурыз, 2014 жыл
- Python 3.5 - 13-ші қыркүйек, 2015 жыл
- Python 3.6 - 23-ші желтоқсан, 2016 жыл
- Python 3.7 - 27-ші шілде, 2018 жыл

Массив

Массив-бір атаумен біріктірілген, бір типтегі деректерден тұратын айнымалылардың жиынтығы. Әрбір алынған жеке айнымалы массив элементі деп аталады. Массив элементтері индекстер арқылы белгіленеді, индекстер тік жақшаның ішінде жазылады және олар нөлден бастап нөмірленеді. Сондықтан да массивтерді кейде индексі бар айнымалы деп атайды.

Бірөлшемді массив – бір атауға біріктірілген бірнеше біртипес элементтерден тұратын жиын. Әрбір элементтің өзінің реттік нөмірі болады. Реттік нөмір – массив элементінің индексі. Массив элементтері бірінен соң бірі ретпен орналасады. Массивті жариялаған кезде массивтің атауын, массив элементтерінің типін көрсетеміз. Массивтің өлшемі көрсетілмейді, ол кез келген уақытта жаңа элементтерді қосу арқылы кеңейтілуі мүмкін. Массивті Python-да пайдалану үшін `array` стандарты модулін импорттау қажет.

Python программасындағы қызметтік сөздер:

'and', 'or', 'in', 'is', 'not', 'if', 'else', 'elif',
'assert', 'break', 'continue', 'def', 'del', 'for',
'from', 'while', 'yield', 'import', 'print', 'return'.

Модуль– функцияларды, нысандарды, кластарды қамтитын дайын программалар. *Мысалы:*

```
import array as arr  
a = arr.array('i'),
```

мұндағы `arr` – `array` модулін әрі қарай пайдалану үшін өзіміз қойған атау;
`a` – қазіргі жағдайдағы массив атауы;
`i` – массив элементінің типі (`integer`).

Массивтің әрбір элементі массив атауы мен оның элементінің индексін көрсету арқылы шақыра аламыз.

Массив элементінің мәні					Массив элементтерінің индексі				
0	1	2	3	4	5	6	7	8	9
7	-5	9	1	0	-2	4	3	6	8

Бірөлшемді массивтермен жұмыс істеу соншалықты қиын емес.

```
# массивті жариялау firstArray  
firstArray = arr.array('i')
```

Массив оның элементтерінің типін көрсету арқылы жарияланады, біздің жағдайда бұл – бүтін (int), содан кейін оныңаты көрсетіледі, бұл – firstArray.

Массивтің 8 элементтен тұрғаны ыңғайлы. Мәндерді өзгерту арқылы массивті инициализациялау керек. Оны массивтің барлық элементтеріне дәйекті түрде өтетін цикл арқылы орындау оңай. $i * 2 - 1$ формуласын қолдана отырып инициализациялаймыз, мұндағы i – массив индексінің мәні.

```
# деректер массивін шығару
```

```
for i in range(0, 8):
```

```
    firstArray.append(i * 2 - 1)
```

Бірөлшемді-массивті-енгізу-және-шығару

Массив элементтерінің мәнін енгізу және шығару үшін параметрі бар цикл қолданылады.

```
#циклде массивтің барлық элементтерін толтыру
```

```
for i in range(0,10):
```

```
    testArray.append(i)
```

```
#массивтің мазмұнын шығару циклі
```

```
for i in range(0,10):
```

```
    print(testArray[i])
```

Массивтер бір-екі айнымалымен жұмыс істеуге болмайтын жағдайларда, яғни бір типті элементтердің белгілі бір тізбегі болған кезде қолданылады. Мысалы, пайдаланушы белгілі бір орынмен, ретпен, басында немесе ортасында ауысуға, олардың мәндерін орындарымен ауыстыруға, өсу/кему және т.б.

бо йынша сұрыптауға тиіс.

Барлығын жақсы игеру үшін төмендегі мысалдың программасын қарастырайық:

```
#Программа пән бойынша қойылған бағаларды сұрайды
point1 = input("Bagany engiz: ")
point2 = input("Bagany engiz: ")
point3 = input("Bagany engiz: ")
point4 = input("Bagany engiz: ")
point5 = input("Bagany engiz: ")
print("Kelesi bagalar engizildi: ")
print(point1+'\t',point2+'\t',point3+'\t',
point4+'\t',point5+'\t')
```

```
C:\Users\Admin\Desktop\Learning\12\bin\Debug\12.exe
Bagany engiz: 5
Bagany engiz: 4
Bagany engiz: 2
Bagany engiz: 5
Bagany engiz: 4

Kelesi bagalar engizildi:
5      4      2      5      4

Process returned 0 (0x0)   execution time : 7.559 s
Press any key to continue.
```

Орындалған программа
нәтижесі

Жоғарыда келтірілген программа пайдаланушыға бес айнымалы бойынша сақталатын бес бағаның нәтижесін сұрайды:

```
point1, point2, point3, point4, point5;
```

Енді осы программаның нұсқасын массивті пайдалану арқылы жасап көрейік:

#программа бойынша қойылған бағаларды сұрайды

#Оларды сақтайды және басып шығарады

```
import array as arr
point = arr.array('i')
for i in range(0,5):
    point.append(int(input("Bagany engiz: ")))
    #inputарқашанstrтмттерінедеректердіқайта-
    #рады, сондықтанint() функ4месысанғатсрлендірушін
    #МАЙДАЛАНЫЛАДЫ.
print("Kelesi bagalar engizildi: ")
for i in range(0,5):
    print(point[i], end='\t')
```

Берілген программа да алғашқы нұсқамен бірдей нәтиже береді, бірақ бұл жерде қысқаша келтірілген. Программалау нұсқасы массивтің қолданылуына байланысты *ыңғайлы* және *ықшамды* болып табылады.

Массивтің типтері

Енді программалауда пайдаланылатын массивтердің негізгі түрлерін қарастырамыз. Массивтер өлшемінің төмендегідей түрлері бар:

- бірөлшемді;
- екіөлшемді немесе матрица;
- үшөлшемді және т.б.

Практикада үшөлшемді массивтер сирек қолданылады. Үшөлшемді массивтерге оқулық немесе кітап мысал бола алады (1-ші индекс – кітаптың беттері десек, 2-ші индекс – беттегі жолдар, ал 3-ші индекс – жолдағы әріптер болады). Бірден көп өлшемі бар массивтерді **көпөлшемді** (екіөлшемді, үшөлшемді массивтер және т.б.) деп атайды.

Python программалау тілінде көпөлшемді классикалық массивтер жоқ. Дегенмен осы платформаның негізгі ерекшеліктері көпөлшемді тізімді (*list*) құруды жеңілдетеді.

Python программалау ортасында массивті жариялау

Массивті жариялау үшін, массивтің атауын және айнымалылардың типін көрсету керек.

Мысалы, массивті `float` типімен жариялайық:

```
import array as arr
a = arr.array('f')
```

Python программалау ортасында массивті инициализациялау

Біз массивті жариялағаннан кейін оны өз программамызда пайдалана аламыз, егер оны инициализацияламаған болсақ (оның элементтеріне бастапқы мән бермесек), оның ешқандай пайдасы болмайды. Массивті инициализация- лауды бірнеше жолмен орындауға болады.

Мысалы, `int` деп жариялағанда:

```
numbers_array = arr.array('i', [7, -5, 9, 1, 0,
-2, 4, 3, 6, 8])
numbers_array
```

массивінде 10 элемент бар. Массивтің өлшемін `len()` функциясын қолдануарқылы табуға болады:

```
len(numbers_array)
```

Егер бастапқыда 9 өлшемді массивтің барлық элементтері бірлік болса, онда циклді қолданамыз:

```
import array as arr
b = arr.array('i')
for i in range(0,9):
b.append(1)
```

Массивті былай да беруге болады:

```
b = arr.array('i', [10, 20, 42])
```

Python программасындағы жүйелілік индексі нөлден басталады. Теріс индекстер элементті кезектің соңынанбастап көрсетеді (–1 – соңғы элемент).*Мысалы*:

```
>>> s = [1, 2, 3, 4, 5]
>>> print s[0], s[2],
s[-1] 1 3 5
```

Бір өлшемді массивтермен жұмыс істегенде, белгілі бір шарттармен элементтерді іздеу қажет болады. Бұл элементтердің жиынтығы арасында ең үлкен немесе ең аз элементті (максимум, минимум) табу, оң немесе теріс, жұп немесе тақ сандардың қосындысын есептеу бар.

Бірөлшемді массивтерге мысалдар

Массивтерді қолданудың мысалдарын практикалық түрде қарастырайық 0–100 аралығынан кездейсоқ алынған.Элементтен тұратын бір өлшемді массивті көрсетіндер.

Нәтижесін есептептендер және шығарындар:

1. Массивтің барлық жұп элементтерінің қосындысы (S1) .
2. Массивтің барлық тақ элементтерінің қосындысы (S2) .
3. Массивтің ең үлкен элементінің мәні (max) .
4. Массивтің ең кіші элементінің мәні (min) .
5. Массивтің саны 30–данкем (Sum) элементтерінің қосындысы.
6. 2–ден 9 ға дейінгі индекстермен берілген массив элементтерінің көбейтіндісі (P) .

Берілген есептердің программасын тиянақты түрде қарастырып, талдаңдар. Алдымен массив элементтерімен жасалған жұмыс нәтижелерін сақтау үшін пайдаланылатын массив пен айнымалыларды анықтаймыз.

```
size = 20
```

```
a = arr.array('i', )
```

```
S1 = 0
```

```
S2 = 0
```

```
Sum = 0
```

```
P = 1
```

Біз массивті жарияладық, енді оған инициализациялауды орындауымыз қажет. Тапсырма бойынша массивті 1-ден 100-ге дейінгі кездейсоқ мәндермен толтыруымыз керек. Енді біздің бірөлшемді массивімізге кездейсоқ мәндерді берейік:

```
for i in range(size):
```

```
    a.append(random.randint(1,100))
```

Әрине, массивтің барлық элементтерін біртіндеп қарастыру үшін және әрқайсысына өзінің мәнін меншіктеуді цикл арқылы жасау өте ыңғайлы. Randint () функциясының 1-ден 100-ге дейінгі аралықта кездейсоқ өлшемдерді реттеу үшін бас тапқы және соңғы мәндерін беруіміз керек. Енді қажетті қосынды мен көбейтіндіні есептейтін негізгі мәселемізге көшейік. Тағы да массив бойынша қарастыру үшін for циклін пайдаланамыз.

```
for i in range(size):
```

```
    if a[i]%2 == 0:
```

```
        S1 += a[i]
```

```
    if a[i]%2 != 0:
```

```
        S2 += a[i]
```

```
    if a[i] < 30:
```

```
        Sum += a[i]
```

```
    if i == 2 or i == 9:
```

```
        P*=a[i]
```

Әдеттегідей барлық 0 мен size аралығындағы бірөлшемді массив элементтерін қарастыру үшін біз for циклін пайдаланамыз, мұндағы size – массив өлшемі. Шарт орындалғанда size қолданылмайды, себебі массив элементі 0-ден басталады. Демек, егер массив 20 элементтен тұратын болса, онда оның индекстері 0-ден 19-ға дейін болады.

1) Одан кейін шарт бойынша массивтің жұп элементтерінің қосындысын табуымыз керек. Оны жасау үшін біз тиісті таңдау шарттарын қолданамыз.

```
if a[i]%2 == 0:
```

```
    S1 += a[i]
```

Мұнда бәрі анық: егер элементтің мәнін 2-ге бөлгенде қалған бөлігі нөлге тең болса, массивтің осы элементінің мәні тең және сәйкесінше ол жалпы қосындыға қосылуы мүмкін. Өйткені біз S1 айнымалысын жалпы соманы жинақтау үшін пайдаланамыз, сонда біз оны дереу инициализацияланған 0 мәнімен жариялаймыз. Бұл $S1 += a[i]$ қысқартылған өрнеккендігін және оның толық формасы $S1 = S1 + a[i]$ болатынын есімізге түсірейік.

2) Әрі қарай шарт бойынша массив элементтерінің барлықтақ мәндерінің қосындысын табу керек. Бұл қосындыны табу үшін кері шарт қолданылады, демек, бөліндінің қалған бөлігі 0-ге тең болмаса, онда қосындының айнымалысы бойынша жиынтықты орындаймыз.

```
if a[i]%2 != 0:
```

```
    S2 += a[i]
```

3–4) Енді ең үлкен элемент (max) және массивтің ең кіші элементін (min) табу мәселелеріне көшеміз.

Алгоритм мынадай:

1) цикл алдында массивтің бірінші элементін minimum және maximum айнымалыларына меншіктеуіміз керек, олар бастапқы ең аз және ең көп мән ретінде сақталады;

2) цикл кезінде бастапқы minimum-нан аз немесе maximum-нан көп мән табылса, онда оларды ауыстырып отырады;

3) максималды және минималды элементтері бар айнымалыларды шығару.

Массивтің әрбір элементін салыстыру керек: егер элементтің ең үлкен мәні айнымалы мәннен үлкен болса, онда мән maximum айнымалыға жазылады; егер элементтің ең аз мәні айнымалы мәннен аз болса, онда мән minimum айнымалыға жазылады: $maximum = a[0]$ //Массивтің 1-ші элементінің мәнін айнымалыға орналастырамыз.

```
minimum = a[0]
```

```
for i in range(size):
```

```
    #егер элементтің мәні max айнымалының мәнінен  
    үлкен болса, онда бұл мәнді айнымалыға жазамыз
```

```
    if a[i] > maximum:
```

```
        maximum = a[i]
```

```
#min-эе дәл солай
```

```
if a[i] < minimum:
```

```
    minimum = a[i]
```

Сондай-ақ Python-да кірістірілген max(), min()

функциялары бар:

```
maximum_1 = max(a)
```

```
minimum_1 = min(a)
```

5) 30-дан аз болатын барлық элементтің қосындысын табу үшін тиістіріктеу шартын пайдаланыңдар.

```
if a[i] < 30:
```

```
    Sum += a[i]
```

6) Соңында біз 2-ші немесе 9-шы индекстегі массив элементтерінің көбейтіндісін табуымыз керек. Егер дәл осы уақытта біз массивтің 2-ші немесе (or) 9-шы элементтерінде тұрған болсақ, онда көбейтіндіні орындаңдар деген шарт береміз.

```
if i == 2 or i == 9:
```

```
    P *= a[i]
```

Төменде осы программаның толық коды көрсетілген: Бірөлшемді массивэлементтерімен жұмыс істеу

Мысалы:

```
import array as arr
```

```
import random
```

```
random.seed(None)
```

```
#қажетті айнымалыларды жариялау
```

```
size = 20
```

```
a = arr.array('i', )
```

```
S1 = 0
```

```
S2 = 0
```

```
Sum = 0
```

```
P = 1
```

```
#массивті экранға шығару функциасы
```

```
def print Array(array):
```

```
for i in range(len(array)):
```

```
    print(array[i], end=" ")
```

```
print('?n')
```

```
#массивті кездейсоқ айнымалы мәндермен толтыру
```

```
for i in range(20):
```

```
    a.append(random.randint(1,100))
```

```
#массивтің мазмұнын экранда басып шығару
```

```
print Array(a)
```

```
#циклде қажетті есептеулерді орындау:
```

```
for i in range(size):
```

```
    if a[i]%2 == 0:
```

```
        S1 += a[i]
```

```
    if a[i]%2 != 0:
```

```

    S2 += a[i]
if a[i] < 30:
    Sum += a[i]
if i == 2 or i==9:
    P *= a[i]
maximum_1 = max(a)
minimum_1 = min(a)
maximum = a[0]
minimum = a[0]
for i in range(len(a)):
    if a[i] > maximum:
        maximum = a[i]
if a[i] < minimum:
    minimum = a[i]
#программаның нәтижелерін көрсету:
print("zhup elementter kosyndysy: ", S1)
print("tak elemetter kosyndysy: ", S2)
print("Min: ", minimum, maximum_1)
print("Max: ", maximum, maximum_1)
print("30 deyingi elementter kosyndysy: ", Sum)
print("2 nemese 9 indexti elementter kobeitindisi: ", P)

```

Бірөлшемді жиымды Паскальда былай жазады:

: array [1..n] of

Мұндағы: – массив(кесте) атауы, 1 алғашқы элемент нөмірі, n- ақырғы элемент нөмірі, -integer, real немесе string...болатын элемент типі, array- (жиым) операторды белгілеуге арналған Паскальдың қызметші сөзі.

Массив элементтерін енгізу

BEGIN

for i:=1 to n do

begin

write(i,'-элементтіенгіз ');

readln(A[i]);

end;

Жиымэлементтерінэкранғашығару

For i:=1 to n do

write (A[i],':2);

Екі өлшемді массивтер

Есептерде деректерді кесте түрінде сақтау жиі кездеседі. Мұндай деректер матрица немесе екі өлшемді массив деп аталады. Python программалау тілінде көпөлшемді массивтер жоқ. Алайда кез келген кесте тізімдер тізімімен ұсынылуы мүмкін. Яғни тізімнің әрбір элементі тізім болып табылады.

Python-дағы тізім кез келген типтегі элементтердің реттелген жиынтығы болуы мүмкін. Массив – бір типтегі элементтердің реттелген жиынтығы, сондықтан тізім массивке қарағанда икемді, бірақ бұл икемділік қарапайым құрылым мен жұмыс істейтін кезде қиындық туғызады. Тізім – динамикалық өзгертін тип, бұл кез келген уақытта тізімге элементтерді қосуға және тізімдегі элементтерді жоюға болады дегенді білдіреді. Екіөлшемді массивті жазу үлгісін қарастырайық: `my_list1 = [[11, 12, 5, 2], [15, 6, 10, 5], [10, 8, 12, 5], [12, 15]]` Көпөлшемді массивте әртүрлі бағандар мен жолдар болуы мүмкін екенін ескеру керек. Екіөлшемді массивтерде әртүрлі типтер болуы мүмкін. `my_list2 = [[11, 12, 5, 2], ['hello', 'hi', 'salem'], [10.1, 8.2, 12.4, 3.5], [12, 15]]` Бұл массив `my_list2`: int, string, float сияқты түрлі деректер типіндегі кірістірілген тізімдерінен тұрады. `my_list2` нәтижесін шығару.



```
C:\Windows\System32\cmd.exe
11 12 5 2
hello hi salem
10.1 8.2 12.4 3.5
12 15
C:\Users\Gulnur\Desktop\Informatics\source>
```

Екіөлшемді массивтердегі мәндерге қолжеткізу. Екіөлшемді массивтердегі деректер элементтеріне екі индекс арқылы қол жеткізуге болады. Бір индекс негізгі массивке, екіншісі ішкі массивте деректер элементінің орналасуына қатысты. Егер тек бір индексті алатын болсақ, онда осы массивтің барлығы индекстің сол позициясы үшін ғана жазылады. Мысал қарастырайық: `print(my_list1[0])` – экранға шығарады `[11, 12, 5, 2]` `print(my_list1[1][2])` – 10-ды экранға шығарады. Толық екіөлшемді массивті басып шығару үшін, біз `for` циклін пайдаланааламыз:

```
for i in myList1:
    for j in i:
        print(j, end=" ")
    print()
```

Келтірілген код мынадай нәтижені береді



```
C:\Windows\System32\cmd.exe
11 12 5 2
15 6 10
10 8 12 5
12 15
C:\Users\Gulnur\Desktop\Informatics\source>
```

Қорытынды сұрақтар

1. Массив деген не?
2. Массив қалай жарияланады? Массив қалай форматталады?
3. Массив қандай деректер типінен тұруы мүмкін?
4. Массив элементтерін белгілі бір параметрлер мен қалай іздеуге болады
5. Max және min массивін табу алгоритмі қалай орындалады?
6. Бірөлшемді массив дегеніміз не?
7. Екіөлшемді массив дегеніміз не?
8. Массив элементтеріне қандай амал қолданамыз?
9. Массив сипаттау үшін қандай сөз тіркесі қолданылады?
10. Ең алғаш рет Python тілін жасауды кім қолға алды?
11. Python-ды қолданатын компаниялар?
12. Ван Руссом Python тілін жасауды қашан қолға алды?
13. Python-ның айнымалы түрі қай кезде анықталады?
14. Python программасындағы қызметтік сөздер?
15. Модуль дегеніміз не?
16. Бірөлшемді жиымды Паскальда қалай жазады?
17. Массивтер өлшемінің неше түрі бар?
18. Үшөлшемді массивге нені айтуға болады?
19. Массивтің типтері?
20. Массивтің ерекшелігі?
21. Python-да тізім мен массивтің қандай айырмашылығы бар?

Python программалау тілінде практикалық тапсырмалар

1 -тапсырма Есеп теулердің нәтижелерін тексеріндер.

```
for i in range(size):  
    S += Array[i]  
for i in range(size):  
    P1 *= Array[i];
```

2-тапсырма.

Хмассивіндегіжүпиндекстіэлементпентақиндекстіэлементтіңорындарыауысатын
дайетіп, олардыкерісіншеорналастырып, кейінумассивінеәкеліпқойындар.

```
import array as arr  
x = arr.array('i')  
print("x:", end="")  
n = 10  
x = input().split()[:n]  
y=x  
for i in range(10):  
    if i % 2 == 0:  
        c = x[i]  
        x[i]=x[i+1]  
        x[i+1]=c  
    y[i] = x[i]  
print("y:", end="")  
for i in range(len(y)):  
    print(y[i], end=" ")
```

3-тапсырма. Берілген массивтегі ең үлкен және ең кіші (maxжәнеmin)
мәндердіауыстырындар.

```
import array as arr  
x = arr.array('i')  
print("x=", end="")  
n = 10  
x = input().split()[:n]  
max = x[0]  
min = x[0]  
for i in range(len(x)):  
    if x[i] > max:  
        max = x[i]  
    elif x[i] < min:  
        min = x[i]  
imax=x.index(max)  
imin=x.index(min)  
print("Max: ", max, "Index of max element:", imax)
```

```

print("Min: ", min, "Index of min element:", imin)
b = x[imin]
x[imin] = x[imax]
x[imax] = b
for i in range(len(x)):
    print(x[i],end=' ')
print()

```

4-тапсырма. n элементтері бар сандық мәндердің бірөлшемдімассиві берілген. Массив элементтерін айналдыра оңға қарай жылжыту керек:

```

A[1] → A[2]; A[2] → A[3]; ... A[n] → A[1]
import random
n=random.randint(1,20)
arr=[random.randint(0,100) for i in range(n)]
print(arr)
for i in range(n):
    arr[-i], arr[-i-1]=arr[-i-1], arr[-i]
arr[0], arr[1]=arr[1], arr[0]
print(arr)

```

5-тапсырма. Python программалау тілінде сұрыптау алгоритмі енгізілген:

```

import array as arr
#n – элементтер саны
n = int(input("Number of elements: "))
mass = arr.array('i')
#массивті толтырамыз
for i in range(n):
    print (i+1, end="-")
    mass.append(int(input("element: ")))
#бастапқы массивті шығару
print ("Initial array: ", end=" ")
for i in range (len(mass)):
    print (mass[i], end=" ")
print ("?n")
#массивті кему ретімен орналастыру
for i in range(n):
    for j in range(n-1):
        if mass[j] < mass[j+1]:
#орын ауыстыру
            temp = mass[j]
            mass[j] = mass[j+1]
            mass[j+1] = temp
#орындалған массивті шығару
print ("Final array: ", end=" ")

```

```
for i in range(len(mass)):
    print(mass[i], end=" ")
```

6-тапсырма. Python программалау тілінің консолдық программасындағы массивті есептеп, нәтижесін көпіршіктепсұрыптау әдісімен экранға шығарудың үлгісі.

```
import array as arr
n = int (input("Number of elements: "))
mass = arr.array('i')
for i in range(n):
    print (i+1, end="-i")
    mass.append(int(input("element: ")))
print("Initial array: ", end=" ")
for i in range(n):
    print(mass[i], end=" ")
print("?n")
noSwap = True
passnum = n-1
while passnum > 0 and noSwap:
    noSwap = False
    for i in range(passnum):
        if mass[i] > mass[i+1]:
            noSwap = True
            temp = mass[i]
            mass[i] = mass[i+1]
            mass[i+1] = temp
    passnum = passnum - 1
print("Final array: ", end=" ")
for i in range(n):
    print(mass[i], end=" ")
```

7-тапсырма. Кестеде Нұр-Сұлтаннан Қазақстанның кейбір қалаларына дейінгі арақашықтық жазылған. Қалаларға дейінгі қашықтық берілген рет бойынша шығатындай, программа құрындар:

$S = \{1224, 196, 1505, 426, 276\}$

ҚР қалалары	Алматы	Қарағанды	Шымкент	Павлодар	Көкшетау
S, км	1224 км	196 км	1505 км	426 км	276 км

8-тапсырма. Еркін берілген массивте элементтердің барлықжұп мәнін жойыңдар.

```
import random
a = []
for i in range(20):
    a.append(random.randint(1,100))
print("A =",a)
b = []
i = 0
while i < len(a):
    if a[i] % 2 == 0:
        b.append(a[i])
        del a[i]
    else:
        i += 1
print("A =",a)
print("B =",b)
```

9-тапсырма. n өлшемді массив және k ($1 \leq k \leq n$) бүтін сан берілген. k реттік нөмрі бар массив элементінің алдына, нөлдік мәні мен жаңа элементті енгізіңдер.

```
import array as arr
n = int(input("Number of elements: "))
k = int(input("Number of elements: "))
mass = arr.array('i')
def printArray(a):
    for i in range(len(a)):
        print(a[i], end=" ")
    print('\n')
for i in range(n):
    print(i+1, end="-i ")
    mass.append(int(input("element: ")))
print("Initial array: ", end = "")
printArray(mass)
mass.insert(k,0)
print("Array: ", end = "")
printArray(mass)
print("Final Array: ", end = "")
printArray(mass)
```

10-тапсырма. Келтірілген мысалда жаңа деректердің элементі 2 индексті позицияға кірістіріледі.

```
from array import *
my_list = [[11, 12, 5, 2], [15, 6,10], [10, 8, 12,
5], [12,15]]
```

```

my_list.insert(2, [0, 5, 11, 13, 6])
for i in my_list:
    for j in i:
        print(j, end=" ")
    print()

```

11-тапсырма. Бұл мысалда массивтегі элементтерді жоюды қарастырамыз.

```

from array import *
myList = [[11, 12, 5, 2], [15, 6, 10], [10, 8, 12,
5], [12, 15]]
del myList[3]
for i in myList:
    for j in i:
        print(j, end=" ")
    print()

```

12-тапсырма. 5x5 матрицасы түрінде екіөлшемді массив берілсін. Жұп бағандағы барлық элемент мәндерін 0 мәнімен, ал тақ бағандағы элемент мәндерін 1 санына ауыстырыңдар.

13-тапсырма. Банкке ақша салу

```

a = input("Enter money amount: ")
years = input("Enter number of years: ")

```

```

def bank(a, years):
    x = 0
    while x != int(years):
        b = (float(a) * 1.1)
        a = b
        x = x + 1
    print(float(a))

```

```

bank(a, years)

```

14-тапсырма. Бірөлшемді жиым элементтерінің ең үлкенін анықтау.

```

Program esep1;
var a:array [1..10] of integer;
max,i:integer;
begin
for i:=1 to 10 do
read(a[i]);
max:=a[9];
for i:=2 to 10 do

```

```
if max  
writeln('еңүлкенэлементі =' ,max);  
end.
```

15-тапсырма. Бірөлшемді жиым элементтерінің ең кішісін анықтау.

```
Program esep1;  
var a:array [1..10] of integer;  
min,i:integer;  
begin  
for i:=1 to 10 do  
read(a[i]);  
min:=a[1];  
for i:=2 to n do  
if min>a[i] then min:=a[i];  
write('еңкішіэлементі =' ,min);  
end.
```

Тест

1. ARRAY операторы қандай типке жатады?

- А) жазбалар
- Б) файл
- В) массив
- С) жиындар
- Д) тип

2. Аралық немесе қорытынды мәліметтерді экранға дискіге енгізу – шығару порттарына жазу.

- А) нәтиже алу (шығару)
- Б) операциялар
- В) амалдар
- С) цикл
- Д) деректер

3. Программада массив өлшемін белгілеуде қандай белгі қолданылады?

- А) []
- Б) { }
- В) ++
- С) (**)
- Д) :=

4. Массив қай түрде сипатталады:

- А) massiv :array of [a..b];
- Б) massiv :array[a-b] of тип;
- В) massiv :тип[a..b];
- С) massiv :array of тип;
- Д) massiv :array[a..b] of тип;

5. Массивтің тақ элементтерінің көбейтіндісін тап.

- А) P=1 for i in range(0,n): if A[i]%2 != 0: P*=A[i].
- Б) P=1 for i in range(0,n): if A[i]%2 == 0: P*=A[i].
- В) P=0 for i in range(0,n): if A[i]%2 != 0: P*=A[i].
- С) P=1 for i in range(0,n): if A[i]%2 == 0: P*=A[i].
- Д) P=1 for i in range(0,n): if A[i]%2 <> 0: P*=A[i].

6. Массивтің жұп элементтерінің қосындысын тап.

- А) S=0 for i in range(0,n): if A[i]%2 == 0: S+=A[i].
- Б) S=0 for i in range(0,n): if A[i]%2 = 0: S+=A[i].
- В) S=1 for i in range(0,n): if A[i]%2 == 0: S+=A[i].
- С) S=0 for i in range(1,n): if A[i]%2 == 0: S+=A[i].
- Д) S=0 for i in range(0,n): if A[2*i] == 0: S+=A[i].

7.Қай жазбада массив элементтерін енгізу көрсетілген?

- A) `a = arr.array('i')`.
- B) `for i in range(10): if A[i]<min: min=A[i]`.
- C) `S=0 for i in range(10): S+=array[i]`.
- D) `for i in range(10): A[i]=random.randint(1,100)`.
- E) `var A: array [1..10] of integer`.

8.Бір атаумен белгіленіп, бір өлшеммен өлшенетін шамалардың реттелген тізбегі не деп атаймыз?

- A) атау
- Б) белгі
- В) жиым
- С) жиын
- Д) тип

9.Динамикалық жиым дегеніміз не?

- A) айнымалы элементтерден тұратын
- Б) элементтері тұрақты болатын жиым
- В) көпөлшемді жиым
- С) бірөлшемді жиым
- Д) барлық жауап дұрыс

10.Массивті баяндау қай түйінді сөзден басталады?

- A) `String;`
- Б) `Record;`
- В) `Array;`
- С) `While;`
- Д) `Goto`

Қорытынды

Бағдарламалау тілдері төменгі деңгейден жылдар өте келе жоғарғы деңгейлі бағдарламалау тілдеріне дейін дамыған. Осы даму барысында не жаңадан тілдер ойланып табылып немесе бұрын болған тілдерді жетілдіру арқылы жаңа бағдарламалау тілдерін жасаған. "Бағдарламалар кітапханасы (Библиотека программ; program library)" — 1) бағдарламалардың белгілі бір жүйемен топтастырылған жиынтығы; 2) жүйеде жұмыс істейтін кез келген адамға қатынас құруға болатын стандартты бағдарламалардың жиынтығы. Бағдарламалар кітапханасы бағдарламашылардың қандай да нақты бір мәселені шешуге арналған қажетті дайын алгоритмдер мен бағдарламаларды таңдауына мүмкіндік жасап, олардың қызметін жеңілдетеді. Python бай стандартты кітапханадан, және модульдердің бай жиынтығынан тұрады. Python мен қосымшалар ең танымал және үлкен фирмалар пайдаланып жазылған, мысалға алып қарайтын болсақ: IBM, Yahoo, Google.com, Hewlett Packard, Infoseek, НАСА, Red Hat, CBS MarketWatch, Microsoft.

Пайдаланылған әдебиеттер:

Златопольский Д.М. Основы программирования на языке Python. М.: ДМК Пресс, 2017. – 284 с.: ил. – ISBN 978-5-97060-552-3.

Электрондық қорлар:

<https://kk.wikipedia.org/wiki/Python>

<https://www.armanpv.com/9-klass?lang=kk>