Жамбыл облысы әкімдігі білім басқармасы

«Білім» кәсіби гуманитарлық-техникалық колледжі

(білім беру ұйымының атауы)

**Оқу сабағының жоспары**

(теориялық немесе өндірістік оқыту)

**Екі өлшемді массивтерімен жұмыс**

(сабақ тақырыбы)

**Модуль/Пән атауы** Бағдарламалық кодты қайта өңдеу

**Дайындаған педагог** Нургисаева У.М

**20\_**25**\_ жылғы** «\_\_\_\_» \_\_\_наурыз\_\_\_

**1. Жалпы мәліметтер**

Курс, оқу жылы, топ 3 курс, 3БҚ-1-22, 3БҚ-2-22

Сабақ түрі: Жаттығу

**2. Мақсаты, міндеттері:**

**Оқу:** Білім алушыларға python (пайтон) программалау тілінде екі өлшемді массивтермен жұмыс, есеп шығаруды жүзеге асыру, қолдану жолдарын ұйымдастыру, программаны өңдеу жұмыснегіздерін үйрету

#### **Дамыту:** Логикалық ойлау қабілеттерін дамыту, есептердің әртүрлі шешімдерін табуға дағдыландыру, программалық кодтың құрылымын дұрыс құру қабілеттерін қалыптастыру.

#### Тәрбиелік: Жауапкершілік пен ұқыптылыққа тәрбиелеу, алгоритмдік ойлауды дамыту, шығармашылық қабілеттерін жетілдіру.

**3. Оқу-жаттығу процесінде білім алушылар меңгеретін күтілетін нәтижелер және кәсіби дағдылар тізбесі:** Python программалау екі өлшемді массивтермен жұмыс, есеп шығаруды жүзеге асыру, қолдану жолдарын ұйымдастыру, программаны өңдеу және пайдалану дағдыларын меңгеру. Join, append, pop, del, remove әдістерінің негіздерін үйренеді, код жазады.

**4. Қажетті ресурстар:** ДК немесе ноутбуктер

Ю.Аляев, О.Козлов. Алгоритмизация и языки программирования  Python, C++, Visual Basic: Учебно-справочное пособие. – М.: Финансы и статистика, 2004

<https://ust.kz/powerpoint/eki_olsemdi_massivter-386835.html>

***5. Сабақтың барысы: (90 минут)***

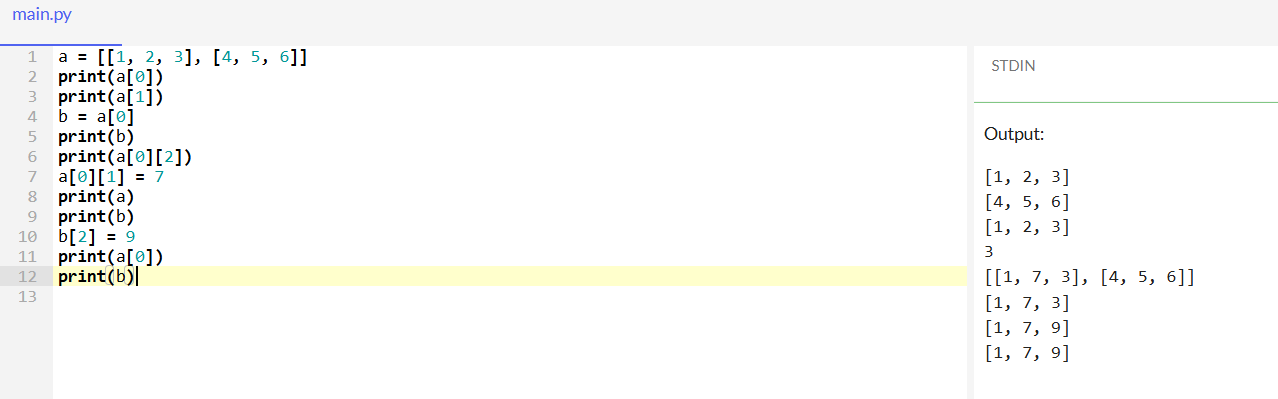
**5.1. Ұйымдастыру кезеңі:** *( 3 мин )*

**5.2. Үй жұмысын жан-жақты тексеру:**

***“Алгоритм” әдісі*** *(15 минут)*

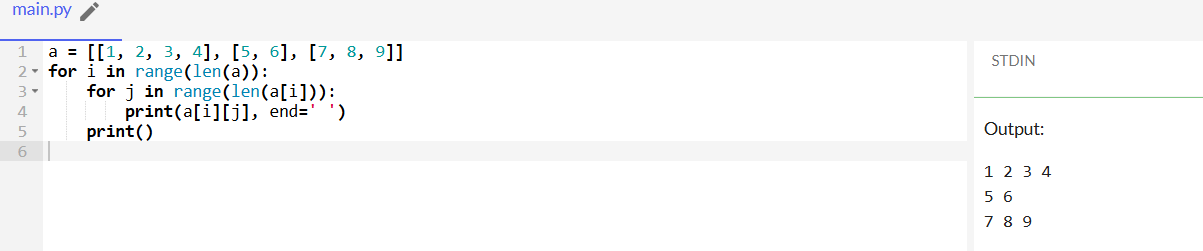
**Екі өлшемді массивтерімен жұмыс**

Көбінесе тапсырмалар деректері бар тікбұрышты кестелерді сақтауды талап етеді. Мұндай кестелер матрицалар немесе екі өлшемді массивтер деп аталады. Python бағдарламалау тілінде кестені жолдар тізімі ретінде көрсетуге болады, оның әрбір элементі өз кезегінде тізім, мысалы, сандар. Мысалы, екі жол мен үш бағаннан тұратын сандық кесте құрылатын, оның көмегімен әртүрлі әрекеттер орындалатын программаны берейік.

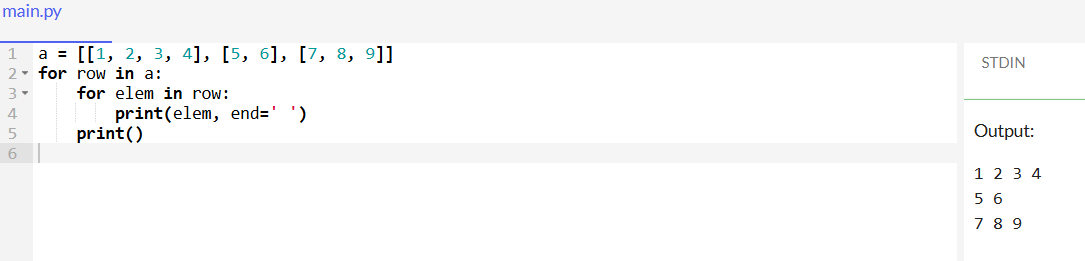


Мұнда a[0] тізімінің бірінші жолы [1, 2, 3] сандар тізімі болып табылады. Яғни, a[0][0] == 1, мән a[0][1] == 2, a[0][2] == 3,a[1][0] == 4, a[ 1 ][1] == 5, a[1][2] == 6.

Тізімді өңдеу және көрсету үшін, әдетте, екі кірістірілген цикл пайдаланылады. Бірінші цикл жол нөмірі бойынша қайталанады, екінші цикл сызық ішіндегі элементтер арқылы өтеді. Мысалы, сандарды бір жолдағы бос орындармен бөліп, екі өлшемді сандық тізімді жол бойынша көрсету үшін мына әрекетті орындауға болады:



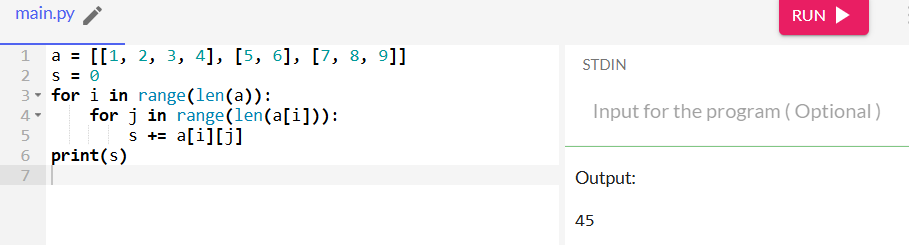
Біз Python тіліндегі for циклінің айнымалысы range() функциясы арқылы жасалған ауқымда ғана емес, сонымен қатар жалпы кез келген тізбектің кез келген элементтерін қайталай алатынын бір рет түсіндіруге тырыстық. Python тіліндегі тізбектер - бұл тізімдер, жолдар және біз әлі кездеспеген басқа да нысандар. Бұл үшін ыңғайлы for циклінің қасиетін пайдаланып екі өлшемді массив шығару жолын көрсетейік:



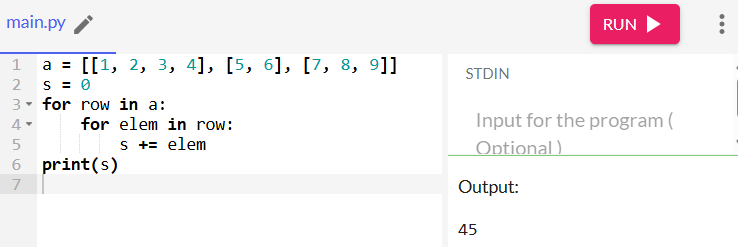
Әрине, бір жолды шығару үшін join() әдісін қолдануға болады:



Тізімдегі барлық сандардың қосындысын есептеу үшін біз екі кірістірілген циклды қолданамыз:



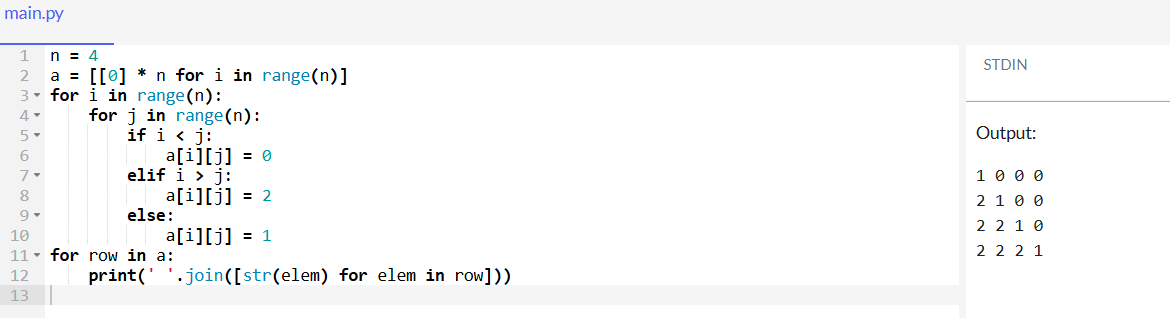
Немесе индекс бойынша емес, жол мәндері бойынша циклмен бірдей нәрсе:



Екі өлшемді массивті өңдеу мысалы

n жолдан және n бағаннан тұратын шаршы массив берейік. Жоғарғы сол жақтан төменгі оң жақ бұрышқа өтетін негізгі диагональда орналасқан элементтерге 1 мәнін беру керек (яғни, a[i][j] элементтері, олар үшін i==j), және негізгі диагональдың үстінде орналасқан элементтерге 0 мәні, негізгі диагональдан төмен элементтердің мәні 2. Яғни, келесі массивді алу керек (n==4 үшін мысал):

Бұл мәселені шешудің бірнеше жолдарын қарастырайық. Негізгі диагоналдың үстінде жатқан элементтер a[i][j] элементтері болып табылады, олар үшін i < j, ал негізгі диагональдан төмен элементтер үшін i>j. Осылайша, біз i және j мәндерін салыстырып, оларды A[i][j] мәнін анықтау үшін пайдалана аламыз. Біз келесі алгоритмді аламыз:



### **1-есеп: Матрицаның бас диагоналының қосындысын табу**

**Тапсырма:**  
Квадрат матрица берілген. Оның бас диагоналындағы (сол жақ жоғарғы бұрыштан оң жақ төменгі бұрышқа дейінгі) элементтердің қосындысын есептеу.

**Шешімі:**

N = int(input())

matrix = []

for i in range(N):

row = list(map(int, input(f"{i+1}-жол элементтерін енгізіңіз: ").split()))

matrix.append(row)

diagonal\_sum = sum(matrix[i][i] for i in range(N))

print("Бас диагональ элементтерінің қосындысы:", diagonal\_sum)

### **2-есеп: Матрицаның ең үлкен және ең кіші элементін табу**

**Тапсырма:**  
Берілген N x M өлшемді матрицадағы ең үлкен (max) және ең кіші (min) элементті табу.

**Шешімі:**

N = int(input())

M = int(input())

matrix = []

for i in range(N):

row = list(map(int, input(f"{i+1}-жол элементтерін енгізіңіз: ").split()))

matrix.append(row)

max\_element = max(max(row) for row in matrix)

min\_element = min(min(row) for row in matrix)

print("Матрицаның ең үлкен элементі:", max\_element)

print("Матрицаның ең кіші элементі:", min\_element)

**3-есеп Екі өлшемді массивті 90 градусқа айналдыру.**

def rotate\_matrix(matrix):

# Массивтің транспозициясын жасау

transposed = list(zip(\*matrix))

# Әр жолды кері тәртіпте айналдыру

rotated = [list(row)[::-1] for row in transposed]

return rotated

matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

rotated\_matrix = rotate\_matrix(matrix)

for row in rotated\_matrix:

print(row)

|  |  |  |
| --- | --- | --- |
| **Білдім** | **Білемін** | **Білгім келеді** |
|  |  |  |

**Рефлексия**

**Үй тапсырмасы –** Зертханалық жұмыс №8 орындау және қорғау