

Матрицы (двумерные массивы)

Создание матриц

! Матрица – это список списков!

```
A = [[-1, 0, 1],  
      [-1, 0, 1],  
      [0, 1, -1]]
```

перенос на другую
строку внутри скобок

или так:

```
A = [[-1, 0, 1], [-1, 0, 1], [0, 1, -1]]
```

! Нумерация элементов с нуля!

Создание вложенных списков

Первый способ: сначала создадим список из n элементов (для начала просто из n нулей). Затем сделаем каждый элемент списка ссылкой на другой одномерный список из m элементов:

```
n = 3
```

```
m = 4
```

```
a = [0] * n
```

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

```
    a[i] = [0] * m
```

Создание вложенных списков

Второй способ: создать пустой список, потом n раз добавить в него новый элемент, являющийся списком-строкой:

```
n = 3
```

```
m = 4
```

```
a = []
```

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

```
    a.append([0] * m)
```

Создание вложенных списков

Третий способ: создать список из n элементов, каждый из которых будет списком, состоящих из m нулей:

```
n = 3
```

```
m = 4
```

```
a = [ [0] * m for i in  
range(n) ]
```

Простые алгоритмы

Заполнение случайными числами:

```
import random
for i in range(N):
    for j in range(M):
        A[i][j] = random.randint(20, 80)
        print("{:4d}".format(A[i][j]),
              end=" ")
    print()
```

Вывод матриц

```
print ( A )
```

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
def printMatrix ( A ) :  
    for row in A :  
        for x in row :  
            print ( "{:4d} " .format ( x ) , end = "" )  
        print ( )
```

```
1    2    3  
4    5    6  
7    8    9
```



Зачем форматный вывод?


```
import random
```

```
#Процедура печати матрицы
```

```
def printMatrix (m):
```

```
    for row in m:
```

```
        for x in row:
```

```
            print ( "{:4d}".format(x) , end = "" )
```

```
        print ()
```

```
n = 3
```

```
m = 4
```

```
a = [[0] * m for i in range(n)]
```

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

```
    for j in range(m):
```

```
        a[i][j] = random.randint ( 20, 80 )
```

```
printMatrix(a)
```

Простые алгоритмы

Суммирование:

```
s = 0
for i in range(N):
    for j in range(M):
        s += A[i][j]
print ( s )
```

```
s = 0
for row in A:
    s += sum(row)
print ( s )
```

Ввод двумерного массива

В первой строке записаны через пробел размеры матрицы: количество строк N и количество столбцов M ($1 \leq N, M \leq 100$). В следующих N строках записаны строки матрицы, в каждой – по M натуральных чисел, разделённых пробелами.

```
n,m = map(int, input().split())
a = []
for i in range(n):
    row = input().split()
    for i in range(len(row)):
        row[i] = int(row[i])
    a.append(row)
```

Ввод двумерного массива

В первой строке записаны через пробел размеры матрицы: количество строк N и количество столбцов M ($1 \leq N, M \leq 100$). В следующих N строках записаны строки матрицы, в каждой – по M натуральных чисел, разделённых пробелами.

```
n,m = map(int, input().split())  
a = []  
for i in range(n):  
    a.append([int(j) for j in  
input().split()])
```

Ввод двумерного массива

В первой строке записаны через пробел размеры матрицы: количество строк N и количество столбцов M ($1 \leq N, M \leq 100$). В следующих N строках записаны строки матрицы, в каждой – по M натуральных чисел, разделённых пробелами.

```
n,m = map(int, input().split())  
a = [[int(j) for j in  
input().split()] for i in range(n)]
```

Задачи

«А»: Напишите программу, которая заполняет квадратную матрицу случайными числами в интервале $[10,99]$, и находит максимальный и минимальный элементы в матрице и их индексы.

Пример:

Матрица А:

12 14 67 45

32 87 45 63

69 45 14 11

40 12 35 15

Максимальный элемент $A[2,2]=87$

Минимальный элемент $A[3,4]=11$

```
import random
n = 3
m = 4
a = [0] * n
for i in range(n):
    a[i] = [0] * m
print(a)
for i in range(n):
    for j in range(m):
        a[i][j] = random.randint (10, 99)
        print ( "{:4d}".format(a[i][j]), end = " " )
    print()
max_i=0
max_j=0
max=a[0][0]
for i in range(n):
    for j in range(m):
        if a[i][j]>max:
            max=a[i][j]
            max_i=i
            max_j=j
print("Максимальный элемент a[" , max_i , " , " , max_j , "]=",
max)
```

Задачи

«В»: Яркости пикселей рисунка закодированы числами от 0 до 255 в виде матрицы. Преобразовать рисунок в черно-белый по следующему алгоритму:

- 1) вычислить среднюю яркость пикселей по всему рисунку
- 2) все пиксели, яркость которых меньше средней, сделать черными (записать код 0), а остальные – белыми (код 255)

Пример:

Матрица А:

12 14 67 45

32 87 45 63

69 45 14 11

40 12 35 15

Средняя яркость 37.88

Результат:

0 0 255 255

0 255 255 255

255 255 0 0

255 0 0 0

Задачи

«С»: Заполните матрицу, содержащую N строк и M столбцов, натуральными числами по спирали и змейкой, как на рисунках:

а)

1	2	3	4
10	11	12	5
9	8	7	6

б)

1	3	4	9
2	5	8	10
6	7	11	12

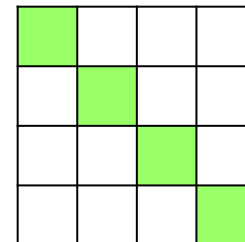
в)

1	6	7	12
2	5	8	11
3	4	9	10

Перебор элементов матрицы

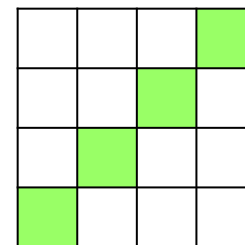
Главная диагональ:

```
for i in range(N):  
    # работаем с A[i][i]
```



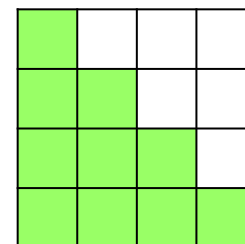
Побочная диагональ:

```
for i in range(N):  
    # работаем с A[i][N-1-i]
```



Главная диагональ и под ней:

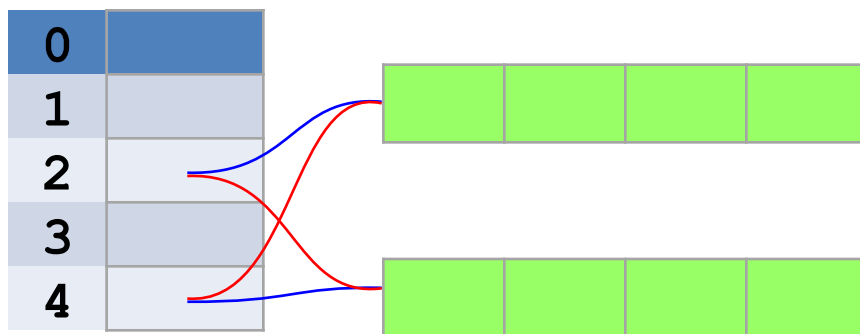
```
for i in range(N):  
    for j in range(i+1):  
        # работаем с A[i][j]
```



Перестановка строк и столбцов

2-я и 4-я строки:

```
A[2], A[4] = A[4], A[2]
```



2-й и 4-й столбцы:

```
for i in range(N):  
    A[i][2], A[i][4] = A[i][4], A[i][2]
```

Выделение строк и столбцов

1-я строка:

```
R = A[1][:]
```

```
R = A[i]
```

2-й столбец:

```
C = []  
for row in A:  
    C.append(row[2])
```

или так:

```
C = [ row[2] for row in A ]
```

главная диагональ:

```
D = [ A[i][i] for i in range(N) ]
```

Задачи

«А»: Напишите программу, которая заполняет квадратную матрицу случайными числами в интервале $[10,99]$, а затем записывает нули во все элементы выше главной диагонали. Алгоритм не должен изменяться при изменении размеров матрицы.

Пример:

Матрица А:

12 14 67 45

32 87 45 63

69 45 14 30

40 12 35 65

Результат:

12 0 0 0

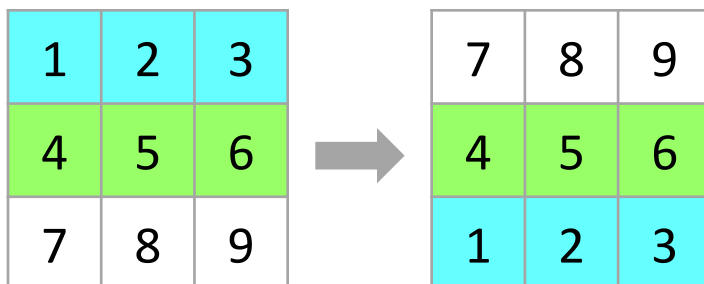
32 87 0 0

69 45 14 0

40 12 35 65

Задачи

«В»: Пиксели рисунка закодированы числами (обозначающими цвет) в виде матрицы, содержащей N строк и M столбцов. Выполните отражение рисунка сверху вниз:



«С»: Пиксели рисунка закодированы числами (обозначающими цвет) в виде матрицы, содержащей N строк и M столбцов. Выполните поворот рисунка вправо на 90 градусов:

