

Двумерные массивы (вложенные списки)

В языке программирования Python таблицу можно представить в виде списка, каждый элемент которого является в свою очередь списком, например, чисел:

A = [[3, 7, 29, 12, 4], [3, 17, 9, 5, 8], [5, 6, 10, 9, 5]]

A[0]
A[1]
A[2]

Индексы двумерного массива:

0,0	0,1	0,2	0,3	0,4	0,5
1,0	1,1	1,2	1,3	1,4	1,5
2,0	2,1	2,2	2,3	2,4	2,5
3,0	3,1	3,2	3,3	3,4	3,5
4,0	4,1	4,2	4,3	4,4	4,5
5,0	5,1	5,2	5,3	5,4	5,5

Главная диагональ

0,0	0,1	0,2	0,3	0,4	0,5
1,0	1,1	1,2	1,3	1,4	1,5
2,0	2,1	2,2	2,3	2,4	2,5
3,0	3,1	3,2	3,3	3,4	3,5
4,0	4,1	4,2	4,3	4,4	4,5
5,0	5,1	5,2	5,3	5,4	5,5

Побочная диагональ

Формулировка задачи	Код
Ввод двумерного массива	
На вход программы подается n, m – количество строк и столбцов, затем сам массив: 3 5 3 4 5 6 3 3 4 5 2 1 1 5 3 2 8	<pre>n, m = map(int, input().split()) A = [] # создаём пустой одномерный массив A for i in range(n): A.append(list(map(int, input().split()))) # считываем n одномерных массивов # и добавляем в исходный массив A</pre>
Предыдущая задача. При считывании массива используем генератор.	<pre>A = [list(map(int, input().split())) for i in range(n)]</pre>
Задача та же, но без использования сложных вызовов функций.	<pre>n, m = map(int, input().split()) A = [] # создаём пустой одномерный массив A for i in range(n): row = input().split() # считываем строку и разбиваем по пробелам for i in range(len(row)): row[i] = int(row[i]) # каждый элемент считанного массива переводим в int A.append(row) # полученный числовой массив добавляем в # исходный массив A</pre>
Создание двумерного массива, состоящего из 0	
Даны два числа: количество строк n и количество столбцов m. Необходимо создать список размером n*m, заполненный нулями.	<pre>A = [0] * n # создаём нулевой одномерный массив for i in range(n): A[i] = [0] * m # каждому элементу исходного списка присваиваем # массив из m элементов</pre>
Задача та же	<pre>A = [] # создаём пустой одномерный массив for i in range(n): A.append([0] * m) # добавляем в исходный массив одномерные массивы, # состоящие из 0</pre>
Создать двумерный нулевой массив с помощью генератора	<pre>A = [[0]*m for i in range(n)]</pre>

Обработка двумерного массива	
Дан массив $n \times n$. Необходимо найти количество элементов, стоящих выше главной диагонали.	<pre>S = 0 for i in range (n): # перебираем строки for j in range (n): # перебираем элементы массива в каждой строке if j > i: # у элементов выше главной диагонали индекс j > индекса i S = S + 1 print(S) # по окончании работы вложенных циклов, выводим S</pre>
Дан массив $n \times m$. Необходимо найти сумму элементов каждой строки.	<pre>for i in range (n): # перебираем строки S = 0 for j in range (m): # перебираем элементы массива в каждой строке S = S + A[i][j] # прибавляем к S очередной элемент массива print(S) # по окончании работы внутреннего цикла, выводим S</pre>
Найти сумму элементов в массиве, организовав перебор самих элементов, не затрагивая индексы.	<pre>S = 0 for row in A: # перебираем строки for elem in row: # перебираем элементы в строке S += elem # прибавляем очередной элемент массива</pre>
Вывод двумерного массива	
Вывести двумерный числовой список $n \times m$ на экран построчно, разделяя числа пробелами внутри одной строки	<pre>for i in range (n): # перебираем строки for j in range (m): # перебираем элементы строки print (A[i][j], end=" ") # выводим элемент строки, затем пробел print () # по окончании вывода строки, переводим курсор # на новую строку</pre>
Вывести двумерный числовой список на экран построчно, разделяя числа пробелами внутри одной строки	<pre>for i in range (len(A)): # перебираем строки в количестве len(A) for j in range (len(A[i])): # перебираем элементы строки в количестве len(A[i]) print (A[i][j], end=" ") # выводим элемент строки, затем пробел print () # по окончании вывода строки, переводим курсор # на новую строку</pre>
Вывести двумерный список, организовав переборы в циклах не по индексу, а по значениям списка	<pre>for row in A: for elem in row: print (elem, end = " ") print ()</pre>
Вывести двумерный список, организовав вывод строк с помощью метода join.	<pre>for row in A: print (" ".join (list (map (str, row))))</pre>