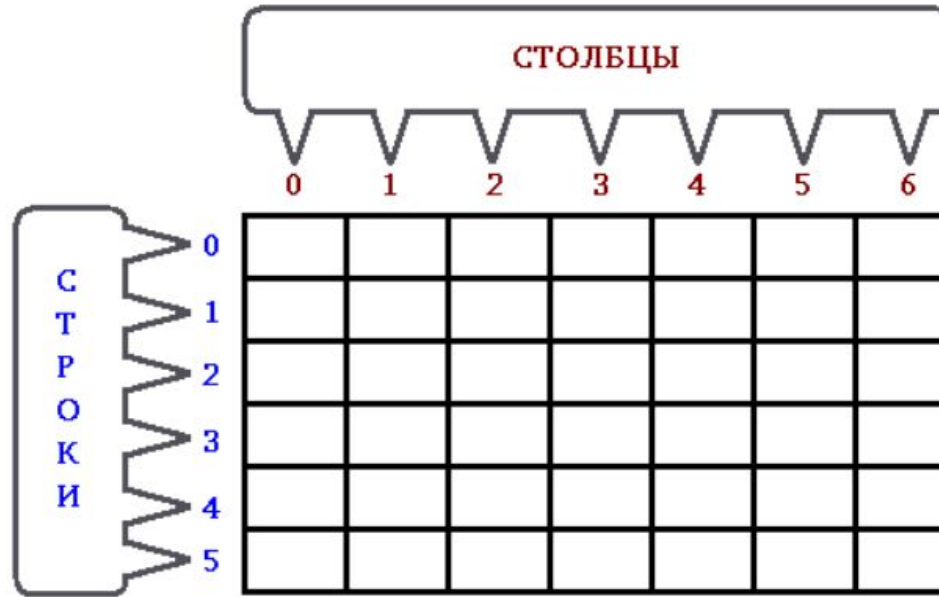


Двумерные массивы.

Что такое двумерный массив?

Двумерный массив это прямоугольная таблица с определенным количеством строк и столбиков. Каждая строка и каждый столбик имеют свой порядковый номер.



В Python подобные таблицы можно представить в виде списка, элементы которого являются другими списками.

Таким образом, получается структура из вложенных списков, количество которых определяет количество строк матрицы, а число элементов внутри каждого вложенного списка указывает на количество столбцов в исходной матрице.

Способы вывода двумерных массивов

Создать двумерный список (с инициализацией или без таковой) не сложно:

Если требуется быстро создать двумерный массив нулевых элементов, то можно воспользоваться структурой вложенных циклов:

Аналогичный результат, но в более компактной форме, можно получить с помощью *генератора*:

```
A = [[], [], [], [], []]
L = [[2, 3, 4], [4, 3, 7], [8, 3, 5]]
```

```
Ar = []
for i in range(10):
    temp = []
    for j in range(10):
        temp.append(0)
    Ar.append(temp)

for i in range(10):
    for j in range(10):
        print(Ar[i][j], end=' ')
    print()
```

```
L = [[0] * 10 for j in range(10)]
print(L)
```

После создания двумерного массива, к его элементу можно получить доступ указав индексы по строке и столбцу.

Например, `Ar[3][2]` означает, что элемент, с которым производится операция, расположен в третьей строке и втором столбце. Таким образом, в отличие от одномерного массива, в двумерном массиве указываются две размерности: первая – **по строкам**, вторая – **по столбцам**.

В ручную матрицы заполнять очень долго, поэтому в тестовых задачах заполнение элементов двумерного массива производится с помощью генератора случайных чисел

```
from random import randint
Ar = [[] for i in range(7)]
for i in range(7):
    for j in range(7):
        Ar[i].append(randint(10,99))
        print('{:<3d}'.format(Ar[i][j]), end='')
print()
```

Построение таблиц истинности при помощи языка

```
print('x y z - F-')
```

- Выводим на экран заголовок

```
for x in range(0, 2):
```

- Задаем значение x, y, z

```
    for y in range(0, 2):
```

```
        for z in range(0, 2):
```

```
            if (_____):
```

- Решение задачи при

```
                print(x, y, z, '-1-')
```

F=1

```
            if not(_____):
```

- Решение задачи при

```
                print(x, y, z, '-0-')
```

F=0

Обозначение операций

\vee - or

\wedge - and

\neg - not()

\equiv - ==

\rightarrow - <=

Построить таблицы истинности

1) $x \vee \neg y \vee z$

2) $x \wedge y \wedge z$

3) $x \wedge y \wedge \neg z$

4) $\neg x \vee y \vee \neg z$