

Двумерные массивы

Иногда данные гораздо удобнее хранить в таблице, чем в столбике/строке. Например, пишем мы игру-лабиринт. Нам нужно как-то хранить карту, где будут отмечены проходы и стены. Например, она могла бы выглядеть вот так.

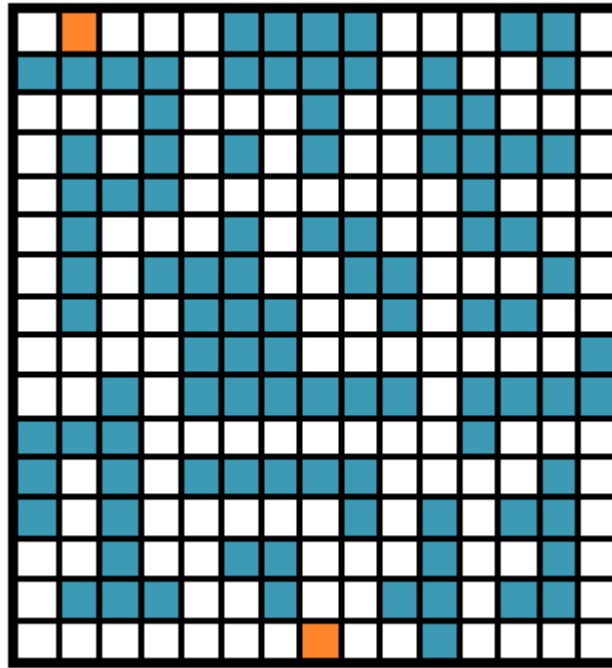


Рис.1 Карта лабиринта

И тогда её можно было бы хранить в виде таблицы символов, где стены обозначены 1, проходы 0, начальная точка 3, конечная точка 9.

Листинг 1.

```
0 9 0 0 0 1 1 1 1 0 0 0 1 1 0
1 1 1 1 0 1 1 1 1 0 1 0 0 1 0
0 0 0 1 0 0 0 1 0 0 1 1 0 0 0
0 1 0 1 0 1 0 1 0 0 1 1 1 1 0
0 1 1 1 0 0 0 0 0 0 0 1 0 0 0
0 1 0 0 0 1 0 1 1 0 0 1 1 0 0
0 1 0 1 1 1 0 0 1 1 0 0 0 1 0
0 1 0 0 1 1 1 0 0 0 0 0 0 0 1
0 0 1 0 1 1 1 1 1 1 0 1 1 1 1
1 1 1 0 0 0 0 0 0 0 0 1 0 0 0
1 0 1 0 1 1 1 1 1 0 0 0 0 1 0
1 0 1 0 0 0 0 0 1 0 1 0 1 1 0
0 0 1 0 0 1 1 0 0 0 1 0 0 1 0
0 1 1 1 0 0 1 0 0 1 1 0 1 1 0
0 0 0 0 0 0 0 3 0 0 1 0 0 0 0
```

Или таблица **12** на **31**, где можно было бы хранить температуру в течение года. Понятно, что в некоторых месяцах будут заполнены не все клетки таблицы, но это ничего страшного. Договоримся и будем туда записывать, например, число **-9999.99**, которое будет обозначать, что данные не определены.

Ещё примеры: таблица Пифагора, поле для игры в крестики нолики, поле для игры **2048**.

В подобных случаях используют двумерные массивы, т.е. массивы, у которых два измерения (два счётчика).

Двумерный массив:

arr2:

	0	1	2	3	4	5	6	7
0	2	3	5	5	3	4	3	4
1	7	1	-3	2	11	243	0	6
2	5	1	-3	5	1	243	-5	-7
3	-4	2	15	2	83	22	2	4

Рис.2 Двумерный массив.

Объявить двумерный массив не сложнее, чем объявить одномерный массив. Просто необходимо указать обе его размерности в квадратных скобках. Первое число – количество строк, второе – количество столбцов.

Например, для массива с картинки объявление выглядело бы вот так:

Листинг 2.

```
int arr2[4][8];
```

Нумерация строк и столбцов, как и в одномерных массивах, начинается с нуля.

Имеется возможность инициализировать массив при объявлении.

Листинг 3.

```
int arr2[4][8] = {0}; // нулями
int arr [2][4] = {{1,2,4,29},{3,4,6,1}}; //произвольными значениями
```

Двумерный массив:

arr:

	0	1	2	3
0	1	2	4	29
1	3	4	6	1

Рис.3 Инициализация двумерного массива произвольными значениями

Пока элементы массива не инициализированы, они заполнены мусором. Проверьте это самостоятельно.

Обращение к конкретному элементу двумерного массива происходит так же, как и в одномерном массиве. Сначала пишется имя массива, а затем в отдельных квадратных скобках номер строки и номер столбца. Главное тут не перепутать. Сначала пишем номер строки, затем номер столбца. Ну и не забывайте про то, что нумерация начинается с нуля.

arr:

	0	1	2	3	4	5	6	7
0	2	3	5	5	3	4	3	4
1	7	1	-3	2	11	243	0	6
2	5	1	-3	5	1	243	-5	-7
3	-4	2	15	2	83	22	2	4

Элемент
arr[0,3]

Элемент
arr[2,4]

Рис.4 Обращение к конкретному элементу двумерного массива.

Следующая программа иллюстрирует основные операции для работы с двумерными массивами.

Листинг 4.

```
#include <stdio.h>
```

```

int main(void) {

    int arr[4][8] = {{2,3,5,5,3,4,3,4},
                     {7,1,-3,2,11,243,0,6},
                     {5,1,-3,5,1,243,-5,-7},
                     {-4,2,15,2,83,22,2,4}};

    printf("%d\t%d\n\n", arr[0][3], arr[2][4]);

    for (int i = 0; i < 4; i = i + 1){
        for(int j = 0; j < 8; j = j + 1){
            printf("%d\t", arr[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```

Как видите, если в одномерном массиве мы использовали один цикл для перебора всех элементов массива, то в двумерном два цикла. Первый цикл последовательно перебирает строки, а второй последовательно перебирает элементы строки. Индексы массива в цикле будут перебираться в следующей последовательности:

```

i = 0;
    j = 0;
    j = 1;
    j = 2;
    j = 3;
    j = 4;
    j = 5;
    j = 6;
    j = 7;
i = 1;
    j = 0;
    j = 1;
    j = 2;
    j = 3;
    j = 4;
    j = 5;
    j = 6;
    j = 7;
и т.д.

```

Следующая программа заполнит массив числами по порядку, а потом выведет его на экран.

Листинг 5.

```

#include <stdio.h>

int main(void) {
    int arr[4][8] = {0};
    int k = 1;

```

```
for (int i = 0; i < 4; i = i + 1){  
    for(int j = 0; j < 8; j = j + 1){  
        arr[i][j] = k;  
        k = k + 1;  
    }  
}  
  
for (int i = 0; i < 4; i = i + 1){  
    for(int j = 0; j < 8; j = j + 1){  
        printf("%d\t", arr[i][j]);  
    }  
    printf("\n");  
}  
  
return 0;  
}
```

Вот и всё, что вам необходимо знать на начальном этапе о массивах. Теперь нужно хорошенько потренироваться в их использовании на практике.

Практика

1. Решите предложенные [задачи](#) с автоматической проверкой решения.

Интернет версия: http://youngcoder.ru/lessons/8/dvumernie_massivy.php