

Передача двумерных массивов в функцию

В функцию можно передавать двумерный массив в качестве параметра, если размер этого массива фиксирован и объявлен в описании функции. То есть если заранее известен размер массива, то можно определить функцию, получающую в качестве параметра двумерный массив такого размера:

```
void f (int A[10][10])
{
    ...
}

int main()
{
    int B[10][10];
    f(B);
}
```

Проблема заключается в том, что в этом случае нельзя использовать массивы произвольного размера.

Чтобы использовать массивы произвольного размера, нам на помощь придут указатели. Для начала разберемся, как представлять двумерный массив в виде указателей.

Одномерный массив `int A[n]` это почти то же самое, что указатель на переменную типа `int`: `int * A`.

Тогда двумерный массив - это массив, каждый из элементов которого является одномерным массивом, то есть указателем на какой-то адрес целого числа в памяти. То есть двумерный массив - это массив элементов типа `int *` или же это указатель на переменную типа `int *`, то есть это переменная типа `int **`.

Итак, двойной указатель можно объявить так:

```
int ** A;
```

Теперь выделим память для массива `A`. Если мы хотим, чтобы в массиве `A` было `n` элементов, каждый из которых является указателем на тип `int`, то сделаем это при помощи операции `new`:

```
A = new int * [n];
```

Теперь `A` указывает на область памяти, содержащей `n` элементов, каждый из которых имеет тип `int *` и указывает на некоторую область памяти, пока еще не выделенную. Выделим эту память - сделаем все `A[i]` указателями на область памяти из `m` элементов типа `int`:

```
for (int i = 0; i < n; ++i)
{
    A[i] = new int [m];
}
```

Функцию, получающую в качестве параметра двумерный массив, можно объявлять так:

```
void f (int ** A, int n, int m)
```

Как и в случае с одномерным массивом, передаваемым как указатель, нам нужно одновременно передавать размеры массива - количество строк `n` и количество столбцов `m`.

При таком способе объявления массива и выделения памяти можно сделать так, чтобы в разных строчках массива было различное число элементов.

Общие требования к оформлению программ.

Считывание данных осуществляется функцией `int ** Read (int & n, int & m)`. Эта функция считывает размер массива в переменные `n` и `m`, передаваемые по ссылке, выделять память под хранение массива, возвращать адрес выделенной памяти.

Решение задачи осуществляется функцией, получающей в качестве параметра массив (типа `int **`), его размеры, дополнительные параметры при необходимости.

Вывод массива на экран осуществляется отдельной функцией `Print`, получающей в качестве параметров массив и его размеры.

Типичный вид программы:

```
void Read (int **& A, int & n)
{
    ...
}

void Fill (int ** A, int n)
{
    ...
}

void Print (int ** A, int n)
{
    ...
}

int main()
{
    int n;
    int ** A;
    Read(A, n);
    Fill(A, n);
    Print(A, n);
    return 0;
}
```

Внимание! Начиная с этого листка я требую от ваших дальнейших программ подобного разбиения на функции.

Плохо структурированные программы приниматься не будут

В. Поменять две диагонали

Дан квадратный массив. Поменяйте местами элементы, стоящие на главной и побочной диагонали, при этом каждый элемент должен остаться в том же столбце (то есть в каждом столбце нужно поменять местами элемент на главной диагонали и на побочной диагонали).

Решение оформите в виде функции `void SwapDiagonals (int ** Src, int n).`

Ввод	Вывод
3	7 2 9
1 2 3	4 5 6
4 5 6	1 8 3
7 8 9	

С. Заполнение змейкой

По данным числам n и m заполните двумерный массив размером $n \times m$ числами от 1 до $n \times m$ “змейкой”, как показано в примере. Выведите полученный массив, отводя на вывод каждого элемента ровно 4 символа.

Ввод	Вывод
3 5	1 2 3 4 5 10 9 8 7 6 11 12 13 14 15

Д. Заполнение диагоналями

По данным числам n и m заполните двумерный массив размером $n \times m$ числами от 1 до $n \times m$ “диагоналями”, как показано в примере. Выведите полученный массив, отводя на вывод каждого элемента ровно 4 символа.

Ввод	Вывод
3 5	1 2 4 7 10 3 5 8 11 13 6 9 12 14 15

Е. Поворот прямоугольного массива

Дан прямоугольный массив размером $n \times m$. Поверните его на 90 градусов по часовой стрелке, записав результат в новый массив размером $m \times n$.

Выведите получившийся массив. Числа при выводе разделяйте одним пробелом.

Ввод	Вывод
3 4 11 12 13 14 21 22 23 24 31 32 33 34	31 21 11 32 22 12 33 23 13 34 24 14

Ф. Поворот квадратного массива

Дан квадратный массив. Поверните его на 90 градусов по часовой стрелке. Результат запишите в этот же массив, вспомогательный массив использовать нельзя.

Выведите результат на экран, разделяя числа одним пробелом.

Ввод	Вывод
3 1 2 3 4 5 6 7 8 9	7 4 1 8 5 2 9 6 3

Г. Седловые элементы

В двумерном массиве $n \times m$, все элементы которого различный, найдите такие элементы, которые одновременно являются минимальными в своей строке и максимальными в своем столбце. Такие элементы называются “седловыми”

Выведите индексы искомых элементов, в одной строке выводите два числа: номер строки и номер столбца, в котором располагается седловой элемент.

Если в массиве нет седловых элементов, выведите одно число 0.

Ввод	Вывод
3 4 1 2 3 4 5 6 7 8 9 10 11 12	2 0
2 2 3 1 2 4	0

Н. Сапер

На поле для игры в сапер клеточки с минами обозначаются символом “*”, а в каждой пустой клеточке записано число от 0 до 8, равное количеству мин в 8 клетках, соседних с данной.

Дан список мин на поле. Постройте по данному списку изображение поля.

Программа получает на вход числа N и M - количество строк и столбцов на поле, а также количество мин на поле K . Далее идет K пар чисел - координат мин. Первое число - номер строки, второе число - номер столбца.

Выведите изображение поля на экран, клетки при выводе разделяйте одним пробелом.

Ввод	Вывод
3 2 2 1 1 2 2	* 2 2 * 1 1
2 2 0	0 0 0 0

I. Заполнение в шахматном порядке

Даны числа n и m . Заполните массив размером $n \times m$ в шахматном порядке: клетки одного цвета заполнены нулями, а другого цвета - заполнены числами натурального ряда сверху вниз, слева направо. В левом верхнем углу записано число 1.

Выведите полученный массив на экран, отводя на вывод каждого элемента ровно 4 символа.

Ввод	Вывод
3 5	1 0 2 0 3 0 4 0 5 0 6 0 7 0 8

J. Заполнение спиралью

По данным числам n и m заполните двумерный массив размером $n \times m$ числами от 1 до $n \times m$ по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в примере. Выведите полученный массив, отводя на вывод каждого элемента ровно 4 символа.

Тесты к этой задаче закрыты.

Ввод	Вывод
4 5	1 2 3 4 5 14 15 16 17 6 13 20 19 18 7 12 11 10 9 8

K. Треугольник Паскаля - 2

Треугольник Паскаля состоит из чисел, где каждое число равно двум числам, стоящим над ним. Если перенумеровать строки треугольника Паскаля с нуля, то i -я строка содержит $i+1$ число, которые равны C_{ji} , где $j \in [0, i]$.

По данному числу n создайте в динамической памяти двумерный массив `int ** A`, выделяя для строки i памяти под хранение $i+1$ элемента i -й строки треугольника Паскаля, то есть

```
A[0] = new int[1];
A[1] = new int[2];
A[2] = new int[3];
```

Далее заполните этот массив числами треугольника Паскаля. Выведите результат на экран отводя на вывод одного числа ровно 6 символов.

Ввод	Вывод					
5	1					
	1	1				
	1	2	1			
	1	3	3	1		
	1	4	6	4	1	