

Двумерные массивы

Задача №354. Побочная диагональ

Числа по модулю не превышают $2^{31}-1$.

Дано число n , $n \leq 100$. Создайте массив $n \times n$ и заполните его по следующему правилу:

- числа на диагонали, идущей из правого верхнего в левый нижний угол, равны 1;
- числа, стоящие выше этой диагонали, равны 0;
- числа, стоящие ниже этой диагонали, равны 2.

Входные данные

Программа получает на вход число n .

Выходные данные

Необходимо вывести полученный массив. Числа разделяйте одним пробелом.

Примеры

входные данные
4
выходные данные
0 0 0 1 0 0 1 2 0 1 2 2 1 2 2 2

main.cpp

Run

Output

```

1 //Побочная диагональ
2 using namespace std;
3 #include <iostream>
4 int main() {
5     int n, matrix[100][100];
6     cin >> n;
7     for (int i = 0; i < n; i++) {
8         for (int j = 0; j < n; j++) {
9             if (i + j == n - 1){
10                 matrix[i][j] = 1;
11             }
12             else{
13                 if (i + j < n - 1){
14                     matrix[i][j] = 0;
15                 }
16                 else {
17                     matrix[i][j] = 2;
18                 }
19             }
20             cout << matrix[i][j] << " ";
21         }
22         cout << endl;
23     }
24
25     return 0;
26 }

```

/tmp/ERP1TmKwYO.o

4

0 0 0 1

0 0 1 2

0 1 2 2

1 2 2 2

=== Code Execution Successful ===

Задача №355. Симметричная ли матрица?

Числа по модулю не превышают $2^{31}-1$.

Проверьте, является ли двумерный массив симметричным относительно главной диагонали. Главная диагональ — та, которая идёт из левого верхнего угла двумерного массива в правый нижний.

Входные данные

Программа получает на вход число $n \ (\leq 100)$, являющееся числом строк и столбцов в массиве. Далее во входном потоке идет n строк по n чисел, являющихся элементами массива.

Выходные данные

Программа должна выводить слово **yes** для симметричного массива и слово **no** для несимметричного.

Примеры

входные данные
3 0 1 2 1 5 3 2 3 4
выходные данные
yes

входные данные
3 0 0 0 0 0 0 1 0 0
выходные данные
no

main.cpp

Run

```
1 //Симметричная ли матрица?
2 #include <iostream>
3 using namespace std;
4 int main() {
5     int n, matrix[100][100];
6     cin >> n;
7     for (int i = 0; i < n; ++i) {
8         for (int j = 0; j < n; ++j) {
9             cin >> matrix[i][j];
10        }
11    }
12    string x = "yes";
13    for (int i = 0; i < n; ++i) {
14        for (int j = 0; j < n; ++j) {
15            if (matrix[i][j] != matrix[j][i]) {
16                x = "no";
17                break;
18            }
19        }
20        if (x == "no") {
21            break;
22        }
23    }
24    cout << x;
25    return 0;
26 }
```

Output

/tmp/wuCkSRgbpq.o
3
0 1 2
1 5 3
2 3 4
yes

=== Code Execution Successful ===

Массивы: Условия задач

[В начало](#) / [Курсы](#) / [Изучение языка программирования](#) / [Массивы](#) / [Тема 2](#) / [Условия задач](#)

Задача №356. Состязания - 1

Числа по модулю не превышают $2^{31}-1$.

В метании молота состязается n спортсменов. Каждый из них сделал m бросков. Победителем считается тот спортсмен, у которого сумма результатов по всем броскам максимальна. Если перенумеровать спортсменов числами от 0 до $n-1$, а попытки каждого из них – от 0 до $m-1$, то на вход программа получает массив $A[n][m]$, состоящий из неотрицательных целых чисел. Программа должна определить максимальную сумму чисел в одной строке и вывести на экран эту сумму и номер строки, для которой достигается эта сумма.

Входные данные

Программа получает на вход два числа n и m , являющиеся числом строк и столбцов в массиве. Далее во входном потоке идет n строк по m чисел, являющихся элементами массива.

Выходные данные

Программа должна вывести 2 числа: сумму и номер строки, для которой эта сумма достигается. Если таких строк несколько, то выводится номер наименьшей из них. Не забудьте, что нумерация строк (спортсменов) начинается с 0.

Примеры

входные данные
2 2 5 4 3 5
выходные данные
9 0

main.cpp

Run

```
1 //Состязания - 1
2 #include <iostream>
3 using namespace std;
4 int main() {
5     int n, m, A[100][100], x = 0, y = 0;
6     cin >> n >> m;
7     for (int i = 0; i < n; ++i) {
8         for (int j = 0; j < m; ++j) {
9             cin >> A[i][j];
10        }
11    }
12    for (int i = 0; i < n; ++i) {
13        int z = 0;
14        for (int j = 0; j < m; ++j) {
15            z += A[i][j];
16        }
17        if (z > x) {
18            x = z;
19            y = i;
20        }
21    }
22    cout << x << endl << y;
23    return 0;
24 }
```

Output

```
/tmp/hrudNeK7Je.o
2 2
5 4
3 5
9
0
=== Code Execution Successful ===
```

Массивы: Условия задач

[В начало](#) / [Курсы](#) / [Изучение языка программирования](#) / [Массивы](#) / [Тема 2](#) / [Условия задач](#)

Задача №357. Состязания - 2

Числа по модулю не превышают $2^{31}-1$.

В метании молота состязается n спортсменов. Каждый из них сделал m бросков. Победителем соревнований объявляется тот спортсмен, у которого максимален наилучший результат по всем броскам. Таким образом, программа должна найти значение максимального элемента в данном массиве, а также его индексы (то есть номер спортсмена и номер попытки).

Входные данные

Программа получает на вход два числа n и m , являющиеся числом строк и столбцов в массиве. Далее во входном потоке идет n строк по m чисел, являющихся элементами массива.

Выходные данные

Программа выводит значение максимального элемента, затем номер строки и номер столбца, в котором он встречается. Если в массиве несколько максимальных элементов, то нужно вывести минимальный номер строки, в которой встречается такой элемент, а если в этой строке таких элементов несколько, то нужно вывести минимальный номер столбца. Не забудьте, что все строки и столбцы нумеруются с 0.

Примеры

входные данные
3 3 3 1 2 1 3 4 3 3 3
выходные данные
4 1 2

main.cpp

Run

Output

```
1 //Состязания - 2
2 #include <iostream>
3 using namespace std;
4 int main() {
5     int n, m, x = -1, y = 0, z = 0;
6     cin >> n >> m;
7     for (int i = 0; i < n; ++i) {
8         for (int j = 0; j < m; ++j) {
9             int f;
10            cin >> f;
11            if (f > x) {
12                x = f;
13                y = i;
14                z = j;
15            } else if (f == x && (i < y || (i == y && j < z))) {
16                y = i;
17                z = j;
18            }
19        }
20    }
21    cout << x << endl << y << " " << z;
22
23    return 0;
24 }
```

/tmp/1jjgCjmhL1.o
3 3
3 1 2
1 3 4
3 3 3
4
1 2

=== Code Execution Successful ===

Массивы: Условия задач

[В начало](#) / [Курсы](#) / [Изучение языка программирования](#) / [Массивы](#) / [Тема 2](#) / [Условия задач](#)

Задача №361. Таблица умножения

Числа по модулю не превышают $2^{31}-1$.

Даны два числа n и m . Создайте двумерный массив $A[n][m]$, заполните его таблицей умножения $A[i][j]=i*j$ и выведите на экран. При этом **нельзя использовать вложенные циклы**, все заполнение массива должно производиться одним циклом.

Входные данные

Программа получает на вход два числа n и m - количество строк и столбцов, соответственно.

Выходные данные

Программа должна вывести полученный массив. Числа разделяйте одним пробелом.

Примеры

входные данные
3 3
выходные данные
0 0 0 0 1 2 0 2 4

main.cpp

Run

```
1 //Таблица умножения
2 #include <iostream>
3 using namespace std;
4 int main() {
5     int n, m, A[100][100];
6     cin >> n >> m;
7     for (int i = 0; i < n * m; ++i) {
8         A[i / m][i % m] = (i / m) * (i % m);
9     }
10    for (int i = 0; i < n; ++i) {
11        for (int j = 0; j < m; ++j) {
12            cout << A[i][j] << " ";
13        }
14        cout << endl;
15    }
16    return 0;
17 }
```

/tmp/ZKjGtEENAC.o

3 3
0 0 0
0 1 2
0 2 4

=== Code Execution Successful ===

Массивы: Условия задач

[В начало](#) / [Курсы](#) / [Изучение языка программирования](#) / [Массивы](#) / [Тема 2](#) / [Условия задач](#)

Задача №362. Треугольник Паскаля

Числа по модулю не превышают $2^{31}-1$.

Даны два числа n и m . Создайте двумерный массив $A[n][m]$ и заполните его по следующим правилам: Числа, стоящие в строке 0 или в столбце 0 равны 1 ($A[0][j]=1, A[i][0]=1$). Для всех остальных элементов массива $A[i][j]=A[i-1][j]+A[i][j-1]$, то есть каждый элемент равен сумме двух элементов, стоящих слева и сверху от него.

Входные данные

Программа получает на вход два числа n и m .

Выходные данные

Выведите данный массив.

Примеры

входные данные		
3	3	
выходные данные		
1	1	1
1	2	3
1	3	6

main.cpp

Run

```
1 //Треугольник Паскаля
2 #include <iostream>
3 using namespace std;
4 int main() {
5     int n, m, A[100][100];
6     cin >> n >> m;
7     for (int i = 0; i < n; ++i) {
8         A[i][0] = 1;
9     }
10    for (int j = 0; j < m; ++j) {
11        A[0][j] = 1;
12    }
13    for (int i = 1; i < n; ++i) {
14        for (int j = 1; j < m; ++j) {
15            A[i][j] = A[i-1][j] + A[i][j-1];
16        }
17    }
18    for (int i = 0; i < n; ++i) {
19        for (int j = 0; j < m; ++j) {
20            cout << A[i][j] << " ";
21        }
22        cout << endl;
23    }
24
25    return 0;
26 }
```

Output

/tmp/XgdhiPfPlv.o
3 3
1 1 1
1 2 3
1 3 6

=== Code Execution Successful ===