

# Двумерные массивы

**Двумерный массив** – структура данных, которая является частным случаем многомерного массива. Двумерный массив – массив одномерных массивов.

Количество размерностей массива стандартом не ограничивается.

Общая форма описания одномерного статического массива:

тип имя\_массива [размер] [размер] ... [размер];

## Пример

Описание двумерного массива целых чисел (тип `int`) с именем `A`.

`int A[2][10];` // 2 – количество строк массива, 10 – количество столбцов массива.

Инициализация массива может выполняться в фигурных скобках после знака равно.

Если число инициализаторов меньше размерности массива, то оставшиеся элементы получают значение 0.

## Пример

Инициализация двумерного массива целых чисел (тип `int`) с именем `A`.

`int A[2][10]={5, -12, -12, 9, 10, 0, -9, -12, -1, 23, 6, 7, 8};` //или

`int A[2][10]={ {5, -12, -12, 9, 10}, {0, -9, -12, -1, 23} };` //или

`int A[ ][10]={ {0}, {0} };`

В многомерных массивах пустой может быть только первая размерность.

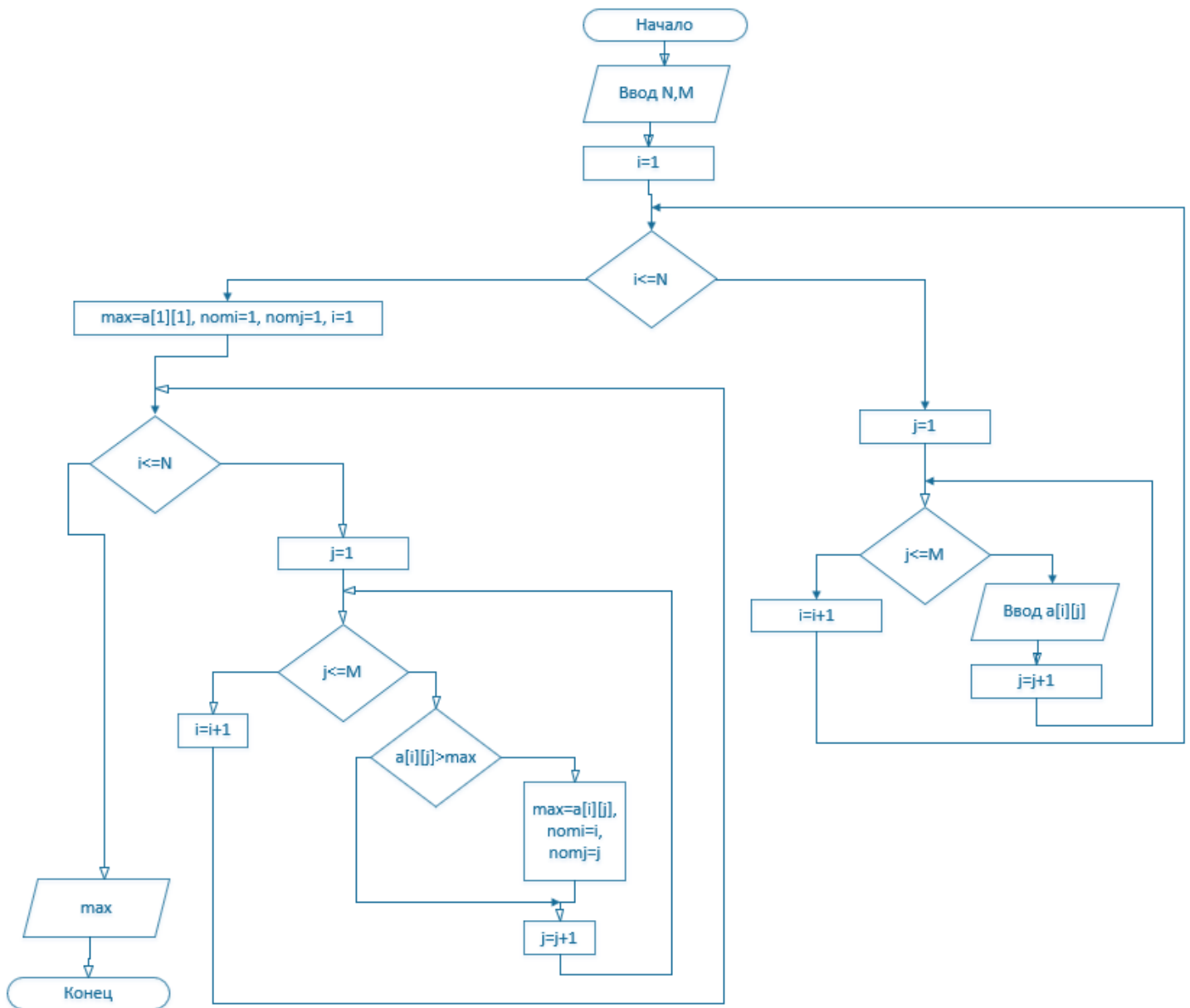
**Пример: Двумерный массив: `a[7][3]`**

<b>a</b>		<b>j – Столбцы</b>		
		→		
↓ <b>i – Строки</b>	<b>i = 0</b>	<b>j = 0</b>	<b>j = 1</b>	<b>j = 2</b>
	<b>i = 1</b>	-100	10	0
	<b>i = 2</b>	12	15	-14
	<b>i = 3</b>	0	-200	46
	<b>i = 4</b>	152	-256	54
	<b>i = 5</b>	83	-25	15
	<b>i = 6</b>	39	27	38
	<b>i = 7</b>	-7	5	93

На пересечении *i*-той строки и *j*-того столбца будет располагаться элемент `a[i][j]` двумерного массива. **На примере:** `a[3][2] = 54`.

## 1. Задача:

Задача: Дана матрица размера  $m \times n$ . Найти максимальный элемент этой матрицы и его индексы.



## 2. Пишем программу:

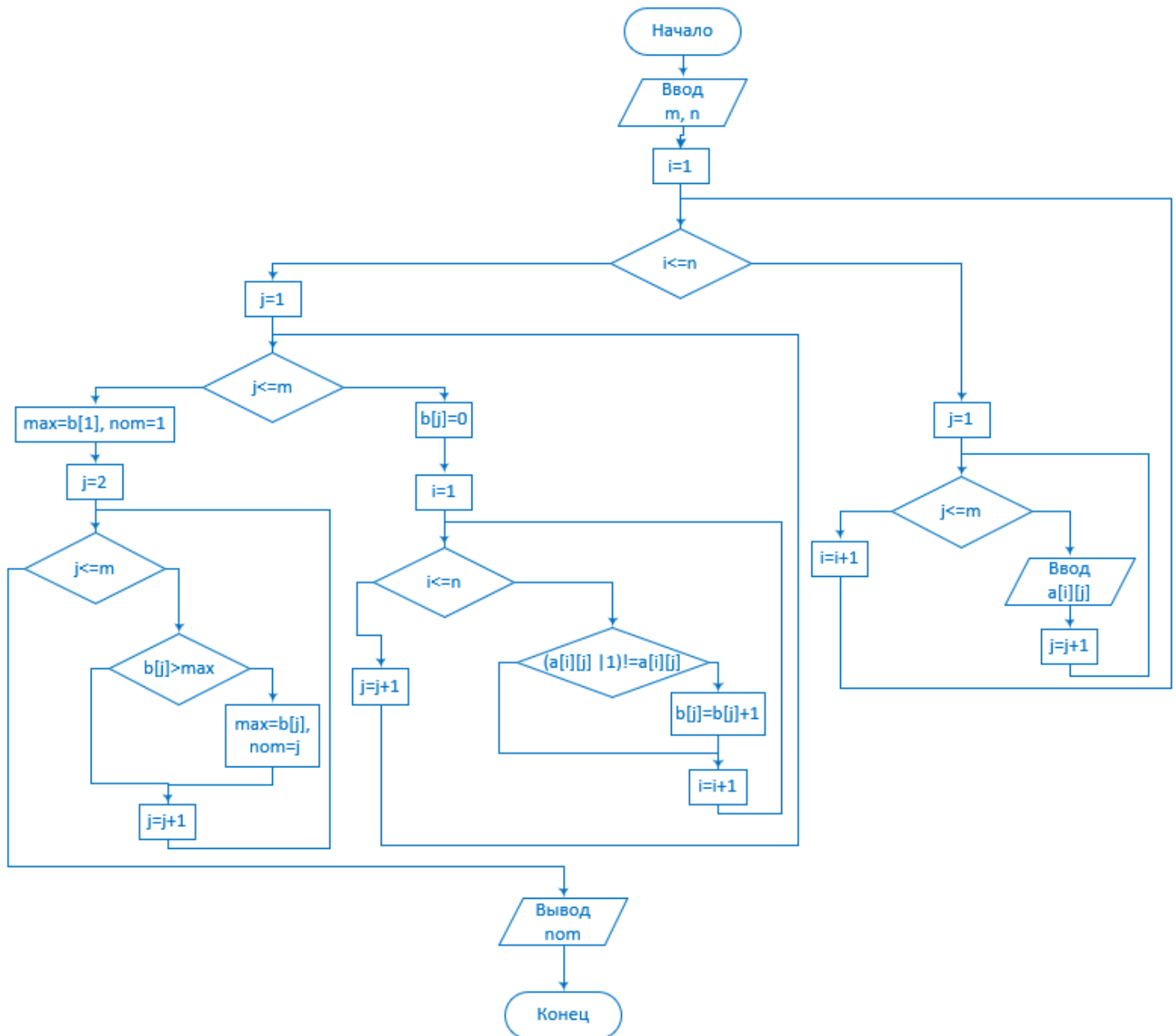
```
#include <stdio.h>
#include <locale.h>
#pragma warning(disable: 4996)

void main(void) {
    int i, j, n, m, nomi, nomj, max, a[20][20];

    printf("Введите n и m\n");
    scanf("%d %d", &n, &m);
    printf("Введите массив построчно\n");
    for(i=1; i<=n; i++)
        for(j=1; j<=m; j++)
            scanf("%d", &a[i][j]);
    max=a[1][1];
    nomi=1;
    nomj=1;
    for(i=1; i<=n; i++)
        for(j=1; j<=m; j++)
            if(a[i][j]>max){
                max=a[i][j];
                nomi=i;
                nomj=j;
            }
    printf("MAX элемент a[%d][%d]=%d\n", nomi, nomj, max);
}
```

### 3. Задача:

Задача: Дана матрица размера  $m \times n$ . Найти номер столбца, в котором находится больше всего четных элементов.



#### 4. Пишем программу:

```
#include <stdio.h>
#include <locale.h>
#pragma warning(disable: 4996)

void main(void) {
    int i, j, n, m, nom, max, a[20][20], b[20];
    printf("Введите n и m\n");
    scanf("%d %d", &n, &m);
    printf("Введите массив построчно\n");
    for(i=1;i<=n;i++)
        for(j=1;j<=m;j++)
            scanf("%d", &a[i][j]);
    for(j=1;j<=m;j++){
        b[j]=0;
        for(i=1;i<=n;i++)
            if((a[i][j]|1)!=a[i][j])
                b[j]++;
    }
    max=b[1];
    nom=1;
    for(j=2;j<=m;j++)
        if(b[j]>max){
            max=b[j];
            nom=j;
        }
    printf("nom=%d\n", nom);
    // построчный вывод массива
    for(i=1;i<=n;i++){
        for(j=1;j<=m;j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
}
```

### **Лабораторная работа №3.**

Для выполнения лабораторной работы необходимо:

1. Получить индивидуальное задание.
2. Составить блок схему алгоритма решения задачи.
3. Написать программу на языке программирования Си в соответствии с алгоритмом, описанным блок схемой.
4. **ОТЛАДИТЬ И ПРОТЕСТИРОВАТЬ ПРОГРАММУ.**
5. Составить отчет о лабораторной работе.

Содержание отчета:

- Титульный лист.
- Условие задачи.
- Блок схема.
- Текст программы.
- Скриншоты, подтверждающие корректную работу программы с разными наборами входных данных (около 5 наборов).