

Бригада №1. (Овчаренко А., Хакимов Д., Куколев К.)

***Документация к программе: поиска
наибольшего/наименьшего элемента в двумерном
массиве.***

Постановка задачи (ПЗ)

Задание: Написать программу обработки двумерного массива в соответствии с условием.

Условие: Найти номер и значение максимального и/или минимального элемента в двумерном массиве array(n,m).

Уточненная ПЗ

Задан двумерный целочисленный массив array, состоящий из n столбцов и m строк. Найти значение первого максимального/минимального элемента min/max и его адрес [row0(строка),column0(столбец)/row1(строка),column1(столбец)] . Элементы матрицы находятся на отрезке [-100,100].

Форма ввода

n
m
<array[1,1]> <array[1,2]> <array[1,m]>
<array[2,1]> <array[2,2]> <array[2,m]>
.....
<array[n,1]> <array[n,2]> <array[n,m]>
variant

Форма вывода

Поиск максимума в массиве

В двумерном массиве <n> строки <m> столбцов

Значения элементов:

<array[1,1]> <array[1,2]> <array[1,m]>
<array[2,1]> <array[2,2]> <array[2,m]>
.....
<array[n,1]> <array[n,2]> <array[n,m]>

Первый минимальный элемент = min и имеет адрес[row0,column0]

Первый максимальный элемент = max и имеет адрес[row1,column1]

Пример работы

n=2,m=3

array:

1 2 3

4 5 6

Поиск максимального:

Максимальный элемент: max = 6, Его адрес: < 2,3 >

Таблица данных

Класс	Имя	Описание (смысл)	Диапазон	Тип
Входные данные	n	количество строк матрицы	$0 < n < 11$	int
	m	количество столбцов матрицы	$0 < m < 11$	int
	array	матрица элементов	$-101 < \text{array}[] < 101$	int
	variant	выбор пути решения	$0 \leq \text{variant} \leq 2$	int
Выходные данные	min	Значение минимального элемента матрицы	$-101 < \text{min} < 101$	int
	max	Значение максимального элемента матрицы	$-101 < \text{max} < 101$	int
	row0	номер строки минимального элемента	$0 < \text{row0} < 11$	int
	column0	номер столбца минимального элемента	$0 < \text{column0} < 11$	int
	row1	номер строки максимального элемента	$0 < \text{row1} < 11$	int
	column1	номер столбца максимального элемента	$0 < \text{column1} < 11$	int
промежуточные	i,j,k,l	вспомогательные переменные прохода по циклу	$0 < i,j,k,l < 11$	int
	A	вспомогательная переменная ввода матрицы	—	string
	f	выявляет выход за границу элементов матрицы	false,true	bool

Тесты:

Общий тест

n = 3

m = 2

2

15	20
3	-7
5	2

Первый минимальный элемент = -7 и имеет адрес[2,2]

Первый максимальный элемент = 20 и имеет адрес[1,2]

Для variant = 2

```

Введите количество строк матрицы: 3
Введите количество столбцов матрицы: 2
Вводите элементы матрицы
15 20
3 -7
5 2

Задана матрица размерами 3 x 2
15      20
3       -7
5       2

Выберите, что вы хотите найти
0 - для поиска минимального элемента
1 - для поиска максимального элемента
2 - для поиска максимального и минимального элементов
2

Первый минимальный элемент = -7 и имеет адрес[2,2]
Первый максимальный элемент = 20 и имеет адрес[1,2]

```

Для variant = 0

```
Введите количество строк матрицы: 3
Введите количество столбцов матрицы: 2
Вводите элементы матрицы
15 20
3 -7
5 2

Задана матрица размерами 3 x 2
15      20
3       -7
5       2

Выберите, что вы хотите найти
0 - для поиска минимального элемента
1 - для поиска максимального элемента
2 - для поиска максимального и минимального элементов
0

Первый минимальный элемент = -7 и имеет адрес[2,2]
```

Для variant = 1

```
Введите количество строк матрицы: 3
Введите количество столбцов матрицы: 2
Вводите элементы матрицы
15 20
3 -7
5 2

Задана матрица размерами 3 x 2
15      20
3       -7
5       2

Выберите, что вы хотите найти
0 - для поиска минимального элемента
1 - для поиска максимального элемента
2 - для поиска максимального и минимального элементов
1

Первый максимальный элемент = 20 и имеет адрес[1,2]
```

n <= 0

n = -1

Неверное значение для количества строк. Укажите значение, принадлежащее отрезку [1;10]

Введите количество строк матрицы: -1
Неверное значение для количества строк. Укажите значение, принадлежащее отрезку [1;10]

m <= 0

n = 2

m = 0

Неверное значение для количества столбцов. Укажите значение, принадлежащее отрезку [1;10]

Введите количество строк матрицы: 2
Введите количество столбцов матрицы: 0
Неверное значение для количества столбцов. Укажите значение, принадлежащее отрезку [1;10]

n > 10

n = 29

Неверное значение для количества строк. Укажите значение, принадлежащее отрезку [1;10]

Введите количество строк матрицы: 29
Неверное значение для количества строк. Укажите значение, принадлежащее отрезку [1;10]

m > 10

n = 2

m = 21

Неверное значение для количества столбцов. Укажите значение, принадлежащее отрезку [1;10]

Введите количество строк матрицы: 2
Введите количество столбцов матрицы: 21
Неверное значение для количества столбцов. Укажите значение, принадлежащее отрезку [1;10]

Несколько минимальных значений → Выводим первое

n = 2

m = 2

variant = 0

10	10
3	3

Первый минимальный элемент = 3 и имеет адрес[2,1].

```

Введите количество строк матрицы: 2
Введите количество столбцов матрицы: 2
Вводите элементы матрицы
10 10
3 3

Задана матрица размерами 2 x 2
10      10
3       3

Выберите, что вы хотите найти
0 - для поиска минимального элемента
1 - для поиска максимального элемента
2 - для поиска максимального и минимального элементов
0

Первый минимальный элемент = 3 и имеет адрес[2,1]

```

$A[i,j] > 100$

n = 2

m = 2

1	2
2200	4

В матрице найден элемент, значение которого не удовлетворяет условию задачи.
Укажите значение в пределах [-100;100]

```

Введите количество строк матрицы: 2
Введите количество столбцов матрицы: 2
Вводите элементы матрицы
1 2
2200 4
В матрице найден элемент, значение которого не удовлетворяет условию задачи. Укажите значение в пределах [-100;100]

```

Несколько максимальных значений → Выводим первое

n = 2

m = 2

10	10
3	3

Первый максимальный элемент = 10 и имеет адрес[1,1].

```

Введите количество строк матрицы: 2
Введите количество столбцов матрицы: 2
Вводите элементы матрицы
10 10
3 3

Задана матрица размерами 2 x 2
10      10
3       3

Выберите, что вы хотите найти
0 - для поиска минимального элемента
1 - для поиска максимального элемента
2 - для поиска максимального и минимального элементов
1

Первый максимальный элемент = 10 и имеет адрес[1,1]

```

$A[i,j] < -100$

n = 2

m = 3

1	2	4
-300	4	7

В матрице найден элемент, значение которого не удовлетворяет условию задачи.
Укажите значение в пределах [-100;100]

```

Введите количество строк матрицы: 2
Введите количество столбцов матрицы: 3
Вводите элементы матрицы
1 2 4
-300 4 7
В матрице найден элемент, значение которого не удовлетворяет условию задачи. Укажите значение в пределах [-100;100]

```

Максимальная нагрузка

n = 10

m = 10

variant = 2

100	...	100
...
100	...	100

Первый минимальный элемент = 100 и имеет адрес[1,1]
Первый максимальный элемент = 100 и имеет адрес[1,1]


```

Введите количество строк матрицы: 10
Введите количество столбцов матрицы: 10
Введите элементы матрицы
100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100

Задана матрица размерами 10 x 10
100    100    100    100    100    100    100    100    100    100
100    100    100    100    100    100    100    100    100    100
100    100    100    100    100    100    100    100    100    100
100    100    100    100    100    100    100    100    100    100
100    100    100    100    100    100    100    100    100    100
100    100    100    100    100    100    100    100    100    100
100    100    100    100    100    100    100    100    100    100
100    100    100    100    100    100    100    100    100    100
100    100    100    100    100    100    100    100    100    100
100    100    100    100    100    100    100    100    100    100

Выберите, что вы хотите найти
0 - для поиска минимального элемента
1 - для поиска максимального элемента
2 - для поиска максимального и минимального элементов
2

Первый минимальный элемент = 100 и имеет адрес[1,1]
Первый максимальный элемент = 100 и имеет адрес[1,1]

```

Минимальная нагрузка

n = 1

m = 1

variant = 0

-100

Матрица состоит из одного элемента = -100. Этот элемент является и максимальным, и минимальным.

```

Введите количество строк матрицы: 1
Введите количество столбцов матрицы: 1
Вводите элементы матрицы
-100

Задана матрица размерами 1 x 1
-100

Выберите, что вы хотите найти
0 - для поиска минимального элемента
1 - для поиска максимального элемента
2 - для поиска максимального и минимального элементов
0

Матрица состоит из одного элемента = -100. Этот элемент является и максимальным, и минимальным

```

Краткое описание логики работы программы.

- 1) Пользователь задаёт n строк с m элементами, которые будем обрабатывать
- 2) Программа узнаёт у пользователя задачу, которая она должна выполнить (поиск минимального / максимального / минимального и максимального)
- 3) За начальный минимальный элемент (\min) и начальный максимальный элемент (\max) берётся элемент матрицы, имеющий адрес $[1, 1]$
- 4) Далее имеем разветвление:
 1. Если $n = 1$ и $m = 1$, то программа выведет альтернативный ответ, в котором укажет, что максимальным и минимальным элементом является значение $\text{array}[1, 1]$
 2. В ином случае, программа выполнит сравнение минимального/максимального значений с каждым элементом матрицы, проходясь в каждой строке по каждому числу и будет присваивать новое значение минимальному и максимальному элементу по правилу (если $\min < \text{элемента}[i, j]$, тогда $\min = \text{элемента}[i, j]$, а также запоминаются строка и столбец, в котором находится это новое минимальное значение / если $\max > \text{элемента}[i, j]$, тогда $\max = \text{элемента}[i, j]$, а также запоминаются строка и столбец, в котором находится это новое максимальное значение)
- 5) Программа выводит пользователю значение \min с его адресом/ \max с его адресом/ \min и \max с их адресами.

Оценка сложности алгоритма.

Вычислительная сложность алгоритма $O(f(N))$ определяется тем, насколько увеличивается количество итераций от количества входных данных.

В нашем случае алгоритм имеет линейную сложность, так как при увеличении объема данных в k раз число итераций тоже увеличивается в k раз.

Код программы

```

using System;

public class Program
{
    public static void Main()
    {
        Console.Write("Введите количество строк матрицы: ");
        int n = Convert.ToInt32(Console.ReadLine());
        if ((n <= 0) || (n > 10))
        {
            Console.WriteLine("Неверное значение для количества строк.  

Укажите значение, принадлежащее отрезку [1;10]");
            return;
        }
        Console.Write("Введите количество столбцов матрицы: ");
        int m = Convert.ToInt32(Console.ReadLine());
        if ((m <= 0) || (m > 10))
        {
            Console.WriteLine("Неверное значение для количества столбцов.  

Укажите значение, принадлежащее отрезку [1;10]");
            return;
        }
        int[,] array = new int[n, m];
        Console.Write("Вводите элементы матрицы\n");
        for (int i = 0; i < n; i++)
        {
            string A = Console.ReadLine();
            int[] curr = A.Split(' ').Select(int.Parse).ToArray(); // преобразование
строки элементов в отдельные int-овые элементы, представление их в виде массива
            for (int j = 0; j < m; j++)
            {
                array[i, j] = curr[j];
            }
        }
        int k = 0, l = 0;
        bool f = true; // флаг имеет значение true, если все элементы
удовлетворяют поставленным условиям
        while ((k < n) && (f))
        {
            l = 0;
            while ((l < m) && (f))
            {
                if ((array[k, l] < -100) || (array[k, l] > 100))
                    f = false;
                l++;
            }
            k++;
        }
    }
}

```

```

        if (f == false)
        {
            Console.WriteLine("В матрице найден элемент, значение которого
не удовлетворяет условию задачи. Укажите значение в пределах [-100;100]");
            return;
        }
        Console.WriteLine("\nЗадана матрица размерами {0} x {1}", n, m);
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < m; j++)
            {
                Console.Write(array[i, j] + " ");
            }
            Console.WriteLine();
        }
        Console.WriteLine("\nВыберите, что вы хотите найти\n0 - для поиска
минимального элемента\n1 - для поиска максимального элемента\n2 - для поиска
максимального и минимального элементов");
        int variant = Convert.ToInt32(Console.ReadLine());
        if ((k == 1) && (l == 1))
        {
            Console.WriteLine("\nМатрица состоит из одного элемента = {0}.
Этот элемент является и максимальным, и минимальным", array[0, 0]);
            return;
        }
        int max = array[0, 0], min = array[0, 0], row0 = 1, column0 = 1, row1 = 1,
column1 = 1;
        k = 0;
        if (variant == 0)
        {
            while (k < n)
            {
                l = 0;
                while (l < m)
                {
                    if (array[k, l] < min)
                    {
                        min = array[k, l];
                        row0 = k + 1;
                        column0 = l + 1;
                    }
                    l++;
                }
                k++;
            }
            Console.WriteLine("\nПервый минимальный элемент = {0} и имеет
адрес[{1},{2}]", min, row0, column0);
        }

```

```

if (variant == 1)
{
    while (k < n)
    {
        l = 0;
        while (l < m)
        {
            if (array[k, l] > max)
            {
                max = array[k, l];
                row1 = k + 1;
                column1 = l + 1;
            }
            l++;
        }
        k++;
    }
    Console.WriteLine("\nПервый максимальный элемент = {0} и имеет
адрес[{1},{2}]", max, row1, column1);
}
if (variant == 2)
{
    while (k < n)
    {
        l = 0;
        while (l < m)
        {
            if (array[k, l] < min)
            {
                min = array[k, l];
                row0 = k + 1;
                column0 = l + 1;
            }
            l++;
        }
        k++;
    }
    k = 0;
    while (k < n)
    {
        l = 0;
        while (l < m)
        {
            if (array[k, l] > max)
            {
                max = array[k, l];
                row1 = k + 1;
                column1 = l + 1;
            }
        }
    }
}

```

```

        }
        l++;
    }
    k++;
}
Console.WriteLine("\nПервый минимальный элемент = {0} и имеет
адрес[{1},{2}]\nПервый максимальный элемент = {3} и имеет адрес[{4},{5}]", min, row0,
column0, max, row1, column1);
}
//int max = array.Cast<int>().Max(); // использовалось бы, если б не нужно
было искать индексы элементов
//int min = array.Cast<int>().Min(); // использовалось бы, если б не нужно
было искать индексы элементов
//Console.Write($"{max}, {min}"); // использовалось бы, если б не нужно
было искать индексы элементов
}
}

```