**Лабораторная работа № 2**

**Тема:** Создание программ обработки массивов.

**Цель работы:** Изучить принципы описания и использования однородных структурированных данных в языке C#, получить практические навыки разработки программ по обработке массивов.

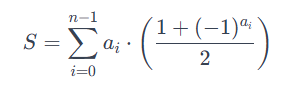
**Задача 1**

**Условие задачи:**

Если в одномерном целочисленном массиве сумма четных элементов положительна, удвоить чётные элементы массива, иначе – удвоить элементы, стоящие после максимального элемента.

**Математическая формулировка:**

Пусть *A*={*a*0​,*a*1​,...,*an*−1​} — одномерный целочисленный массив размерностью *n*.

Вычислить сумму всех четных элементов массива:



Где, это функция, которая равна 1, если *ai*​ четное, и 0, если *ai*​

нечетное.

1. Если *S*>0:
   * Удвоить все четные элементы массива:
2. Иначе (если *S*≤0):
   * Найти индекс максимального элемента *k*
   * Удвоить все элементы, стоящие после максимального элемента:

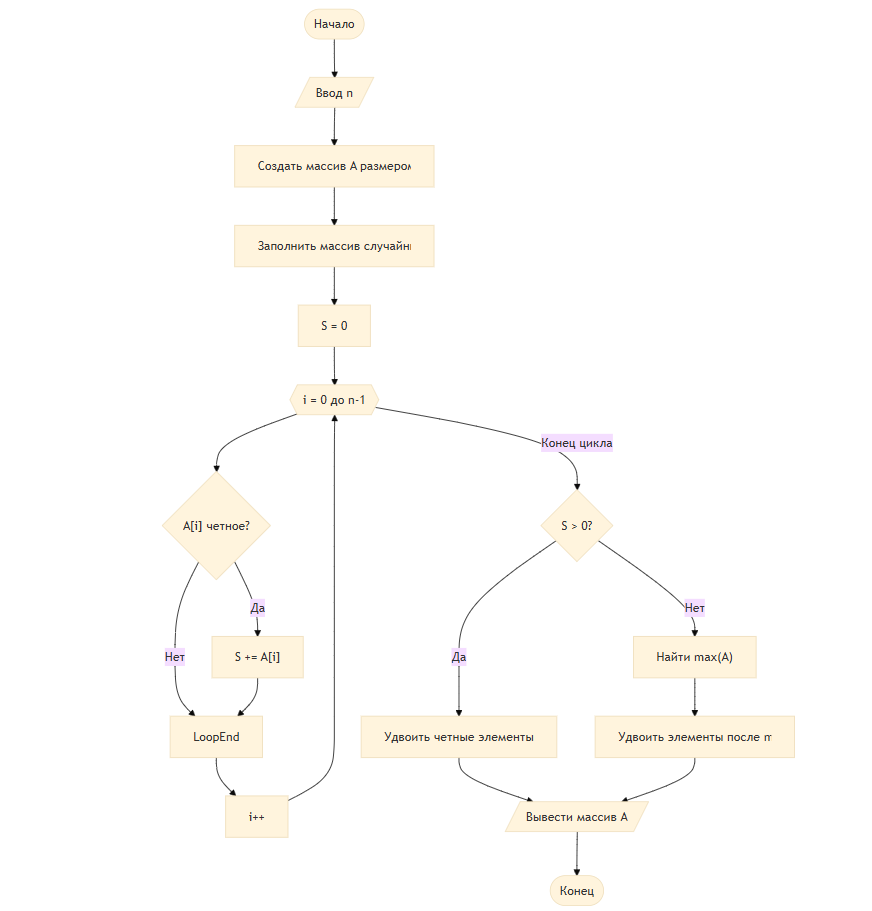
**Ограничения на входные данные:**

* Размерность массива *n* должна быть больше 0.
* Элементы массива могут быть любыми целыми числами.

**Варианты результатов:**

* Массив с удвоенными четными элементами, если сумма четных элементов положительна.
* Массив с удвоенными элементами, стоящими после максимального элемента, если сумма четных элементов не положительна.

**Алгоритм функции** Main:



**Код программы с комментариями:**

using System;

namespace ArrayProcessing

{

    class Program

    {

        static void Main(string[] args)

        {

            // Ввод размерности массива

            Console.Write("Введите размерность массива: ");

            int n = int.Parse(Console.ReadLine());

            // Создание и заполнение массива случайными числами

            int[] array = new int[n];

            Random rnd = new Random();

            for (int i = 0; i < n; i++)

            {

                array[i] = rnd.Next(-1000, 1000);

                Console.Write("{0,8}", array[i]);

            }

            Console.WriteLine();

            // Вычисление суммы четных элементов

            int sumEven = 0;

            foreach (int num in array)

            {

                if (num % 2 == 0)

                {

                    sumEven += num;

                }

            }

            // Проверка условия и выполнение соответствующего действия

            if (sumEven > 0)

            {

                // Удвоение четных элементов

                for (int i = 0; i < n; i++)

                {

                    if (array[i] % 2 == 0)

                    {

                        array[i] \*= 2;

                    }

                }

            }

            else

            {

                // Поиск максимального элемента и его индекса

                int maxIndex = 0;

                for (int i = 1; i < n; i++)

                {

                    if (array[i] > array[maxIndex])

                    {

                        maxIndex = i;

                    }

                }

                // Удвоение элементов после максимального

                for (int i = maxIndex + 1; i < n; i++)

                {

                    array[i] \*= 2;

                }

            }

            // Вывод измененного массива

            Console.WriteLine("Измененный массив:");

            foreach (int num in array)

            {

                Console.Write("{0,8}", num);

            }

            Console.WriteLine();

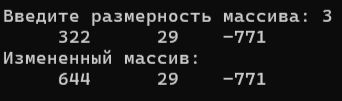
        }

    }

}

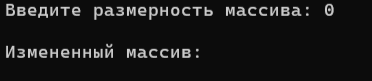
**Тестовые примеры и скриншоты выполнения программ:**

**Работа программы:**



* **Входные данные:** Размерность массива **n = 3**.
* **Сгенерированный массив: 322, 29, -771**.
* **Результат:** Сумма четных элементов положительна = **644**, поэтому четные элементы удваиваются: **644, 29, -771**.

**Ввод нуля:**



* **Входные данные:** Размерность массива **n = 0**.
* **Сгенерированный массив:**
* **Результат:** Сумма четных элементов положительна (**0+ 0 = 0**), поэтому четные элементы удваиваются:

**Задача 2**

**Условие задачи:**

Вывести на экран двумерный массив, раскрасив в нем зеленым цветом первый положительный элемент, синим цветом — последний положительный элемент.

**Математическая формулировка:**

Пусть *B*={*bij*​} — двумерный целочисленный массив размерностью *m*×*n*.

1. Найти координаты первого положительного элемента
2. Найти координаты последнего положительного элемента
3. Вывести массив *B* с раскрашенными элементами:
   * Зелёным цветом выделить элемент первый положительный элемент массива​​.
   * Синим цветом выделить последний положительный элемент *массива*​​.

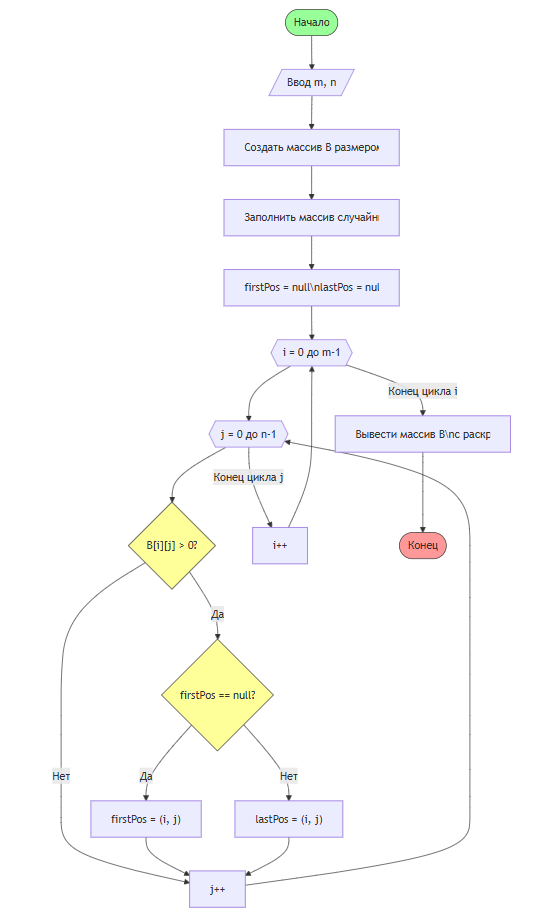
**Ограничения на входные данные:**

* Размерности массива *m* и *n* должны быть больше 0.
* Элементы массива могут быть любыми целыми числами.

**Варианты результатов:**

* Двумерный массив с выделенными первым и последним положительными элементами.

**Алгоритм, построенный MS Visio:**

****

**Код программы с комментариями:**

using System;

namespace ArrayProcessing

{

    class Program

    {

        static void Main(string[] args)

        {

            // Ввод размерностей массива

            Console.Write("Введите количество строк: ");

            int m = int.Parse(Console.ReadLine());

            Console.Write("Введите количество столбцов: ");

            int n = int.Parse(Console.ReadLine());

            // Создание и заполнение массива случайными числами

            int[,] matrix = new int[m, n];

            Random rnd = new Random();

            for (int i = 0; i < m; i++)

            {

                for (int j = 0; j < n; j++)

                {

                    matrix[i, j] = rnd.Next(-1000, 1000);

                    Console.Write("{0,8}", matrix[i, j]);

                }

                Console.WriteLine();

            }

            // Поиск первого и последнего положительных элементов

            Tuple<int, int> firstPos = null;

            Tuple<int, int> lastPos = null;

            for (int i = 0; i < m; i++)

            {

                for (int j = 0; j < n; j++)

                {

                    if (matrix[i, j] > 0)

                    {

                        if (firstPos == null)

                        {

                            firstPos = Tuple.Create(i, j);

                        }

                        lastPos = Tuple.Create(i, j);

                    }

                }

            }

            // Вывод массива с раскрашенными элементами

            Console.WriteLine("Массив с раскрашенными элементами:");

            for (int i = 0; i < m; i++)

            {

                for (int j = 0; j < n; j++)

                {

                    if (firstPos != null && i == firstPos.Item1 && j == firstPos.Item2)

                    {

                        Console.ForegroundColor = ConsoleColor.Green;

                        Console.Write("{0,8}", matrix[i, j]);

                        Console.ResetColor();

                    }

                    else if (lastPos != null && i == lastPos.Item1 && j == lastPos.Item2)

                    {

                        Console.ForegroundColor = ConsoleColor.Blue;

                        Console.Write("{0,8}", matrix[i, j]);

                        Console.ResetColor();

                    }

                    else

                    {

                        Console.Write("{0,8}", matrix[i, j]);

                    }

                }

                Console.WriteLine();

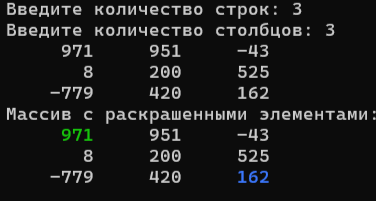
            }

        }

    }

}

**Тестовые примеры и скриншоты выполнения программ:**

****

**Тестовый пример для Задания 2:**

* **Входные данные:** Количество строк **m = 3**, количество столбцов **n = 3**.
* **Сгенерированный массив:**

971 951 -43

8 200 525

- 779 420 162

* **Результат:** Первый положительный элемент **971** раскрашен зеленым, последний положительный элемент **162** раскрашен синим.