

**Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики**

ЛАБОРАТОРНАЯ РАБОТА №5

Создание и использование массивов

Студент:
Швалов Даниил Андреевич
Факультет ИКТ
Группа: K32211
Преподаватель:
Иванов Сергей Евгеньевич

Упражнение 1. Работа с массивом размерного типа данных

В первой части задания необходимо создать массив, занулить все четные элементы и вывести элементы массива в стандартный вывод:

```
namespace Loop;
class Program
{
    static void Main(string[] args)
    {
        int[] myArray = { 100, 1, 32, 3, 14, 25, 6, 17, 8, 99 };

        for (int i = 0; i < myArray.Length; ++i)
        {
            if (myArray[i] % 2 == 0)
            {
                myArray[i] = 0;
            }
            Console.Write("{0} ", myArray[i]);
        }
    }
}
```



Terminal - Loop

0 1 0 3 0 25 0 17 0 99

Рис. 1.1: Вывод программы

Во второй части задания необходимо считать массив со стандартного входа и вывести значения введенного массива в стандартный вывод:

```
namespace Loop;
class Program
{
    static void Main(string[] args)
    {
        int[] myArray;

        Console.Write("Enter array size: ");
        int n = int.Parse(Console.ReadLine());
        myArray = new int[n];

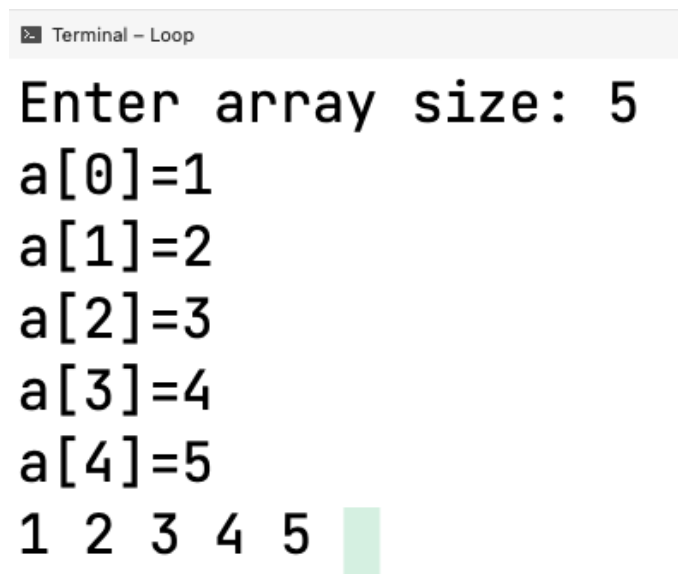
        for (int i = 0; i < myArray.Length; ++i)
        {
            Console.Write("a[{0}]= ", i);
        }
    }
}
```

```

        myArray[i] = int.Parse(Console.ReadLine());
    }

    foreach (int x in myArray)
    {
        Console.Write("{0} ", x);
    }
}

```



```

Terminal - Loop

Enter array size: 5
a[0]=1
a[1]=2
a[2]=3
a[3]=4
a[4]=5
1 2 3 4 5

```

Рис. 1.2: Пример работы программы

Упражнение 2. Перемножение матриц

Сначала реализуем перемножение матриц на заранее заданных данных:

```

namespace MatrixMultiply;
class MatrixMultiply
{
    static void Main(string[] args)
    {
        int[,] a = new int[2, 2];
        a[0, 0] = 1;
        a[0, 1] = 2;
        a[1, 0] = 3;
        a[1, 1] = 4;

        int[,] b = new int[2, 2];
        b[0, 0] = 5;
        b[0, 1] = 6;
        b[1, 0] = 7;
    }
}

```

```

        b[1, 1] = 8;

        int[,] result = new int[2, 2];
        result[0, 0] = a[0, 0] * b[0, 0] + a[0, 1] * b[1, 0];
        result[0, 1] = a[0, 0] * b[0, 1] + a[0, 1] * b[1, 1];
        result[1, 0] = a[1, 0] * b[0, 0] + a[1, 1] * b[1, 0];
        result[1, 1] = a[1, 0] * b[0, 1] + a[1, 1] * b[1, 1];

        Console.WriteLine(result[0, 0]);
        Console.WriteLine(result[0, 1]);
        Console.WriteLine(result[1, 0]);
        Console.WriteLine(result[1, 1]);
    }
}

```

 Terminal – MatrixMultiply

19
22
43
50

Рис. 2.1: Результат работы программы

Выделим вывод массива `result` в отдельную функцию `Output`:

```

namespace MatrixMultiply;
class MatrixMultiply
{
    private static void Output(int[,] result)
    {
        for (int r = 0; r < result.GetLength(0); r++)
        {
            for (int c = 0; c < result.GetLength(1); c++)
            {
                Console.Write("{0} ", result[r, c]);
            }
        }
    }
}

```

```

    }
    Console.WriteLine();
}

static void Main(string[] args)
{
    int[,] a = new int[2, 2];
    a[0, 0] = 1;
    a[0, 1] = 2;
    a[1, 0] = 3;
    a[1, 1] = 4;

    int[,] b = new int[2, 2];
    b[0, 0] = 5;
    b[0, 1] = 6;
    b[1, 0] = 7;
    b[1, 1] = 8;

    int[,] result = new int[2, 2];
    result[0, 0] = a[0, 0] * b[0, 0] + a[0, 1] * b[1, 0];
    result[0, 1] = a[0, 0] * b[0, 1] + a[0, 1] * b[1, 1];
    result[1, 0] = a[1, 0] * b[0, 0] + a[1, 1] * b[1, 0];
    result[1, 1] = a[1, 0] * b[0, 1] + a[1, 1] * b[1, 1];

    Output(result);
}
}

```

 Terminal – MatrixMultiply

19 22
43 50

Рис. 2.2: Результат работы программы

Выделим расчет значений массива в отдельную функцию Multiply:

```

namespace MatrixMultiply;
class MatrixMultiply

```

```

{
    private static void Output(int[,] result)
    {
        for (int r = 0; r < result.GetLength(0); r++)
        {
            for (int c = 0; c < result.GetLength(1); c++)
            {
                Console.Write("{0} ", result[r, c]);
            }
            Console.WriteLine();
        }
    }

    private static int[,] Multiply(int[,] a, int[,] b)
    {
        int[,] result = new int[2, 2];
        result[0, 0] = a[0, 0] * b[0, 0] + a[0, 1] * b[1, 0];
        result[0, 1] = a[0, 0] * b[0, 1] + a[0, 1] * b[1, 1];
        result[1, 0] = a[1, 0] * b[0, 0] + a[1, 1] * b[1, 0];
        result[1, 1] = a[1, 0] * b[0, 1] + a[1, 1] * b[1, 1];
        return result;
    }

    static void Main(string[] args)
    {
        int[,] a = new int[2, 2];
        a[0, 0] = 1;
        a[0, 1] = 2;
        a[1, 0] = 3;
        a[1, 1] = 4;

        int[,] b = new int[2, 2];
        b[0, 0] = 5;
        b[0, 1] = 6;
        b[1, 0] = 7;
        b[1, 1] = 8;

        int[,] result = Multiply(a, b);
        Output(result);
    }
}

```

Изменим тело функции Multiply, заменив текущее решение на решение с циклом for:

```

namespace MatrixMultiply;
class MatrixMultiply
{
    private static void Output(int[,] result)

```

```

{
    for (int r = 0; r < result.GetLength(0); r++)
    {
        for (int c = 0; c < result.GetLength(1); c++)
        {
            Console.Write("{0} ", result[r, c]);
        }
        Console.WriteLine();
    }
}

private static int[,] Multiply(int[,] a, int[,] b)
{
    int[,] result = new int[2, 2];
    for (int r = 0; r < result.GetLength(0); r++)
    {
        for (int c = 0; c < result.GetLength(1); c++)
        {
            result[r, c] += a[r, 0] * b[0, c] + a[r, 1] * b[1, c];
        }
    }
    return result;
}

static void Main(string[] args)
{
    int[,] a = new int[2, 2];
    a[0, 0] = 1;
    a[0, 1] = 2;
    a[1, 0] = 3;
    a[1, 1] = 4;

    int[,] b = new int[2, 2];
    b[0, 0] = 5;
    b[0, 1] = 6;
    b[1, 0] = 7;
    b[1, 1] = 8;

    int[,] result = Multiply(a, b);
    Output(result);
}
}

```

Добавим возможность считывания матриц:

```

namespace MatrixMultiply;
class MatrixMultiply
{
    private static void Output(int[,] result)
    {

```

```

        for (int r = 0; r < result.GetLength(0); r++)
        {
            for (int c = 0; c < result.GetLength(1); c++)
            {
                Console.Write("{0} ", result[r, c]);
            }
            Console.WriteLine();
        }
    }

    private static int[,] Multiply(int[,] a, int[,] b)
    {
        int[,] result = new int[2, 2];
        for (int r = 0; r < result.GetLength(0); r++)
        {
            for (int c = 0; c < result.GetLength(1); c++)
            {
                result[r, c] += a[r, 0] * b[0, c] + a[r, 1] * b[1, c];
            }
        }
        return result;
    }

    private static void Input(int[,] a)
    {
        for (int r = 0; r < a.GetLength(0); r++)
        {
            for (int c = 0; c < a.GetLength(1); c++)
            {
                Console.Write("Enter value for [{0},{1}] : ", r, c);
                string s = System.Console.ReadLine();
                a[r, c] = int.Parse(s);
            }
        }
        Console.WriteLine();
    }

    static void Main(string[] args)
    {
        int[,] a = new int[2, 2];
        Console.WriteLine("Enter a matrix values");
        Input(a);

        int[,] b = new int[2, 2];
        Console.WriteLine("Enter b matrix values");
        Input(b);

        int[,] result = Multiply(a, b);
        Output(result);
    }
}

```



```
Enter a matrix values
Enter value for [0,0] : 1
Enter value for [0,1] : 2
Enter value for [1,0] : 3
Enter value for [1,1] : 4

Enter b matrix values
Enter value for [0,0] : 5
Enter value for [0,1] : 6
Enter value for [1,0] : 7
Enter value for [1,1] : 8

19 22
43 50
```

Рис. 2.3: Результат работы программы

Упражнение 3. Обработка данных массива

Каждую запрашиваемую функциональность реализуем в отдельных функциях. Исходный код программы представлен на следующем листинге:

```
namespace MatrixOperations;
class MatrixOperations
{
    private static double[] ReadArray()
    {
        Console.Write("Enter array size: ");
        int size = int.Parse(Console.ReadLine());
        double[] array = new double[size];

        for (int i = 0; i < size; i++)
        {
            Console.Write("a[{0}]= ", i);
            array[i] = double.Parse(Console.ReadLine());
        }

        return array;
    }
}
```

```

}

private static double Sum(double[] array)
{
    double total = 0;
    foreach (double value in array)
    {
        total += value;
    }
    return total;
}

private static double Average(double[] array)
{
    return Sum(array) / array.Length;
}

private static double SumPositive(double[] array)
{
    double total = 0;
    foreach (double value in array)
    {
        if (value > 0)
        {
            total += value;
        }
    }
    return total;
}

private static double SumNegative(double[] array)
{
    double total = 0;
    foreach (double value in array)
    {
        if (value < 0)
        {
            total += value;
        }
    }
    return total;
}

private static double SumEven(double[] array)
{
    double total = 0;
    for (int i = 0; i < array.Length; i += 2)
    {
        total += array[i];
    }
    return total;
}

```

```

private static double SumOdd(double[] array)
{
    double total = 0;
    for (int i = 1; i < array.Length; i += 2)
    {
        total += array[i];
    }
    return total;
}

private static int MaxIndex(double[] array)
{
    int maxIndex = 0;
    for (int i = 1; i < array.Length; ++i)
    {
        if (array[i] > array[maxIndex])
        {
            maxIndex = i;
        }
    }
    return maxIndex;
}

private static int MinIndex(double[] array)
{
    int minIndex = 0;
    for (int i = 1; i < array.Length; ++i)
    {
        if (array[i] < array[minIndex])
        {
            minIndex = i;
        }
    }
    return minIndex;
}

private static double MultiplyBetweenMinMax(double[] array)
{
    int minIndex = MinIndex(array);
    int maxIndex = MaxIndex(array);
    int start = Math.Min(minIndex, maxIndex) + 1;
    int end = Math.Max(minIndex, maxIndex) - 1;

    if (start > end)
    {
        return 0;
    }

    double total = 1;
    for (int i = start; i ≤ end; ++i)
    {

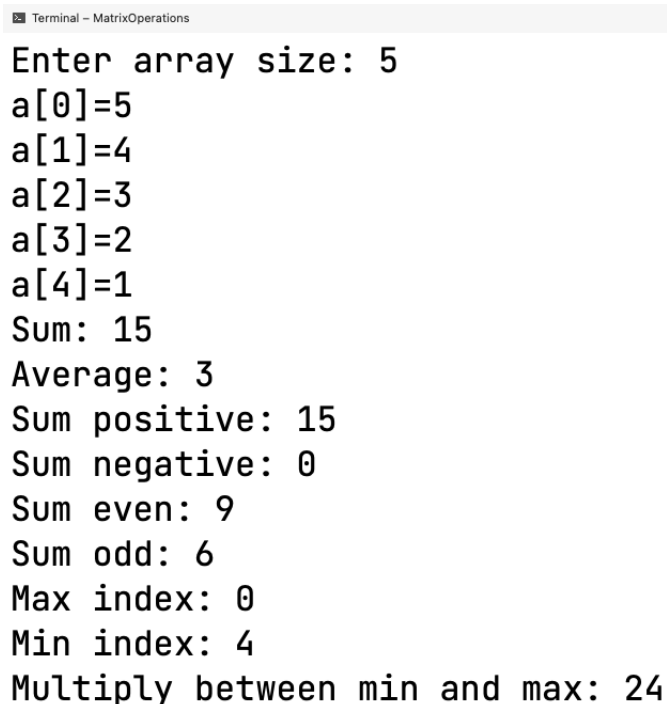
```

```

        total *= array[i];
    }
    return total;
}

static void Main(string[] args)
{
    double[] array = ReadArray();
    Console.WriteLine("Sum: {0}", Sum(array));
    Console.WriteLine("Average: {0}", Average(array));
    Console.WriteLine("Sum positive: {0}", SumPositive(array));
    Console.WriteLine("Sum negative: {0}", SumNegative(array));
    Console.WriteLine("Sum even: {0}", SumEven(array));
    Console.WriteLine("Sum odd: {0}", SumOdd(array));
    Console.WriteLine("Max index: {0}", MaxIndex(array));
    Console.WriteLine("Min index: {0}", MinIndex(array));
    Console.WriteLine(
        "Multiply between min and max: {0}", MultiplyBetweenMinMax(array)
    );
}
}

```



```

Terminal - MatrixOperations
Enter array size: 5
a[0]=5
a[1]=4
a[2]=3
a[3]=2
a[4]=1
Sum: 15
Average: 3
Sum positive: 15
Sum negative: 0
Sum even: 9
Sum odd: 6
Max index: 0
Min index: 4
Multiply between min and max: 24

```

Рис. 3.1: Пример работы программы