Министерство науки и высшего образования РФ
ФГАОУ ВПО
Национальный исследовательский технологический университет «МИСиС»

Институт Информационных технологий и компьютерных наук (ИТКН)

Кафедра Инфокоммуникационных технологий (ИКТ)

Отчет по лабораторной работе №3
по дисциплине «Программирование и Алгоритмизация»
на тему «Массивы. Типовые алгоритмы обработки массивов»

Выполнил:
студент группы БИВТ-24-5

Черных Богдан

Проверил:
Стучилин В. В.

Москва, 2024

Теоретическое введение. Массив – это структура данных, содержащая несколько значений одного типа, обозначаемая одним именем. Доступ к элементам массива осуществляется по индексу. Изменяя индексы, можно переходить от одного элемента массива к другому и таким образом обрабатывать единообразно большие наборы данных, используя циклы. Индексация массивов в C# начинается с нуля. Массив может быть одномерным, многомерным, вложенным. Здесь будут рассмотрены только одномерные и двухмерные массивы. Одномерный массив представляет собой линейную структуру. Положение элемента определяется одним индексом. Двухмерный массив можно представить себе как таблицу. Положение элемента определяется двумя индексами: номером строки и номером столбца. Начнем рассмотрение с одномерных массивов.

Цель работы: освоение фундаментальных концепций работы с одномерными и двумерными массивами в языке программирования C#. Работа направлена на приобретение практических навыков создания, инициализации, доступа к элементам и обработки массивов. Основные задачи включают реализацию таких операций, как суммирование элементов, поиск максимальных и минимальных значений, удаление и вставка элементов, сортировка и перестановка. Особое внимание уделяется применению типовых алгоритмов обработки массивов, использованию циклов для эффективной работы с большими наборами данных, а также применению встроенных методов и свойств класса Array. Конечная цель - научиться разрабатывать программы различной сложности для решения задач, связанных с обработкой массивов в C#.

Код требуемых заданий (1 уровень 12 задание, 1 уровень 13 задание, 1 уровень 14 задание, 2 уровень 12 задание, 2 уровень 13 задание, 2 уровень 14 задание, уровень 12 задание, 2 уровень 13 задание).

(В коде используется 1 checker = 21 только для удобства – из-за того, что я использую одни и те же название переменных, например a(array), если бы не базовое условие(if), мне бы пришлось каждый раз сбрасывать значение, или код бы у меня просто некорректно работал. Поэтому перед каждой новой задачей я пишу if (checker == 21)).
Код:

```
//svg does precious
using System;                       // Аналог <iostream> для работы с консолью и основными функциями
using System.Collections.Generic;   // Аналог <vector>, <list>, <map>, <set>, <unordered_map>, <unordered_set>, <stack>, <queue>
```

```
using System.Text;                    // Аналог <string>, <cstring> (для работы со строками и StringBuilder)
using System.Linq;                    // Аналог <algorithm> (для работы с LINQ, сортировок, поиска и т.д.)
using System.IO;                      // Аналог <cstdio>, <fstream> (работа с файлами)
using System.Globalization;           // Аналог <iomanip> (для форматирования)
using System.Collections;             // Работа с различными коллекциями (например, ArrayList)
using System.Threading;               // Потоки и многопоточность
using System.Runtime.Serialization;   // Аналог <stdexcept> (работа с исключениями)
using System.Reflection;              // Аналог <typeinfo> (информация о типах, рефлексия)
using System.Diagnostics;             // Аналог <utility>, <std::pair> (вспомогательные функции и классы)
using System.ComponentModel;          // Дополнительные утилиты и атрибуты
using System.Numerics;                // Работа с большими числами и математическими операциями
using System.Globalization;
using System.Diagnostics;
using System.Net;
using System.Numerics;
// Для работы с потоками данных:
using System.Threading.Tasks;         // Асинхронные задачи

// Для работы с датами и временем:
using System.Timers;                  // Для работы с таймерами и временем
using System.Collections.Generic;
using System.Text;
using System.Linq;
using System.IO; //important
using System.Globalization;
using System.Collections;
using System.Threading;
using System.Runtime.Serialization;
using System.Reflection;
using System.Diagnostics;
using System.ComponentModel;
using System.Numerics;
using System.Globalization;
using System.Diagnostics;
using System.Net;
```

```
using System.Numerics;
using System.Threading.Tasks;
using System.Security.Cryptography;
using System.Data;
using System.Data.SqlClient;
using System.Xml;
using System.Xml.Linq;
using System.Runtime.InteropServices;
using System.Security;
using System.Web;
using System.Media;
using System.Drawing;
using System.Configuration;
using System.Timers;
using System.Runtime.Remoting;
using System.Runtime.CompilerServices;
using System.Runtime.Versioning;
using System.CodeDom;
using System.CodeDom.Compiler;
using System.Collections.Concurrent;
using System.Runtime;
using System.Windows;
using System.Windows.Input;
using System.Security.Principal;
using System.Security.Permissions;
using System.Resources;
using C = System.Console; //console
using dl = System.Decimal;//decimal
using str = System.String;//string
using l = System.Int64;   //long
using u = System.UInt64;  //Ulong
using db = System.Double; //Double

/*
Izzspot - 19 years
Boogie B - 20 years
SJ - 21 years
Bandokay - life sentence

Youngest in the charge
OFB, we don't window shop
Bro caught him an opp and tried turn him off (Bow, bow)
```

```
In this X3, man's swervin' off (Skrr, skrr)
Free Boogie Bando, he got birded off (Free my bro, free my bro)
Whenever we get a burner loss
We just cop a next one and go burst it off (Ay)
Lil bro's tellin' me he got his earnings wrong
We just took him OT, now his trapline's gone (Ring, ring)
Hashtag
Bro backed this ting and just started squeezin' (Clarted)
When it broad day, it was freezin'
Hashtag fuckery, hashtag screamin'
One on the hand ting woi, left man leanin', leanin' (Fucker)
Show us cause it's good to feel it
Shortie's cooze and she must be dreamin'
*/
/* ¡Adelante Barcelona, adelante Cataluña! Visca el Barça! Visca Catalunya!
   ¡Al diablo con todos los demás, porque lo más importante en la vida es el fútbol!
*/
//--------------------------------------------------------------------------------
//--------------------------------------------------------------------------------
//--------------------------------------------------------------------------------

class program
{//Лабораторная работа №3

    //indexes go from 0, so i bring this, not 1, if element
    //on 7 place he 'll have index value equal to 6

    // for 13 and 14 lvl1
    static void svg(int[] arr) // function for output
    {
        foreach (int num in arr)
        {
            Console.Write($"{num} ");
        }
        Console.WriteLine();
    }
    // for 13 and 14 lvl1

    static void Main()
    {
        int checker = 21;
        if (checker == 21)
```

```
{//Lvl 1, ex 12
   C.WriteLine("Level 1 Exercise 12");
   int[] a = new int[8];
   Random rnd = new Random();
   for (int i = 0; i < a.Length; i++)
   {
      a[i] = rnd.Next(-200, 200);
   }
   Console.WriteLine("Исходный массив:");
   Console.WriteLine(string.Join(" ", a));
   int lastNegativeIndex = -1;
   int lastNegativeValue = 0;
   for (int i = 0; i < a.Length; i++)
   {
      if (a[i] < 0) //Negative
      {
         lastNegativeIndex = i;
         lastNegativeValue = a[i];
      }
   }
   if (lastNegativeIndex != -1) Console.WriteLine($"Последний отрицательный элемент: значение = {lastNegativeValue}, номер = {lastNegativeIndex}");
   else Console.WriteLine("В массиве нет отрицательных элементов");
}


if (checker == 21)
{//Lvl 1, ex 13

   C.WriteLine("Level 1 Exercise 13");
   int[] a = new int[10];
   int[] a02468 = new int[5];
   int[] a13579 = new int[5];

   Random rnd = new Random();
   for (int i = 0; i < a.Length; i++)
   {
      a[i] = rnd.Next(-300, 300);
   }
   svg(a);
   for (int i = 0; i < a.Length; i++)
```

```
        {
            if ((i % 2) == 0)
            {
                a13579[i / 2] = a[i];
            }
            if ((i % 2) != 0)
            {
                a02468[i / 2] = a[i];
            }
        }
        svg(a13579);
        svg(a02468);
    }


    if (checker == 21)
    {//Lvl 1, ex 14
        C.WriteLine("Level 1 Exercise 14");
        int[] a = new int[11];
        Random rnd = new Random();
        for (int i = 0; i < a.Length; i++)
        {
            a[i] = rnd.Next(-400, 400);
        }
        Console.WriteLine("Исходный массив:");
        svg(a);
        int sumOfSquares = 0;
        bool foundNegative = false;
        for (int i = 0; i < a.Length; i++)
        {
            if (a[i] < 0)
            {
                foundNegative = true;
                break;
            }
            sumOfSquares += a[i] * a[i];
        }
        if (foundNegative)
        {
            Console.WriteLine($"Сумма квадратов элементов до первого
отрицательного: {sumOfSquares}");
        }
```

```
            else
            {
                Console.WriteLine("В массиве нет отрицательных элементов");
            }
        }


        if (checker == 21)
        {//lvl 2, ex 12
            C.WriteLine("Level 2 Exercise 12");
            int[] a = new int[10]; // based = 10, cause in task no clue what size
            Random rnd = new Random();
            for (int i = 0; i < a.Length; i++)
            {
                a[i] = rnd.Next(-500, 500);
            }
            Console.WriteLine("Исходный массив:");
            Console.WriteLine(string.Join(" ", a));

            int maxIndex = Array.IndexOf(a, a.Max());
            int firstNegativeIndex = Array.FindIndex(a, x => x < 0);

            if (firstNegativeIndex != -1 && maxIndex != a.Length - 1)
            {
                int sumAfterMax = a.Skip(maxIndex + 1).Sum();
                // вычисляет сумму всех элементов массива a, которые находятся после максимального элемента
                a[firstNegativeIndex] = sumAfterMax;
                // строка заменяет значение первого отрицательного элемента в массиве a на сумму, вычисленную в предыдущей строке.
                Console.WriteLine("Массив после замены:");
                Console.WriteLine(string.Join(" ", a));
            }
            else
            {
                Console.WriteLine("Невозможно выполнить замену: нет отрицательных элементов или максимальный элемент последний");
            }
        }


        if (checker == 21)
```

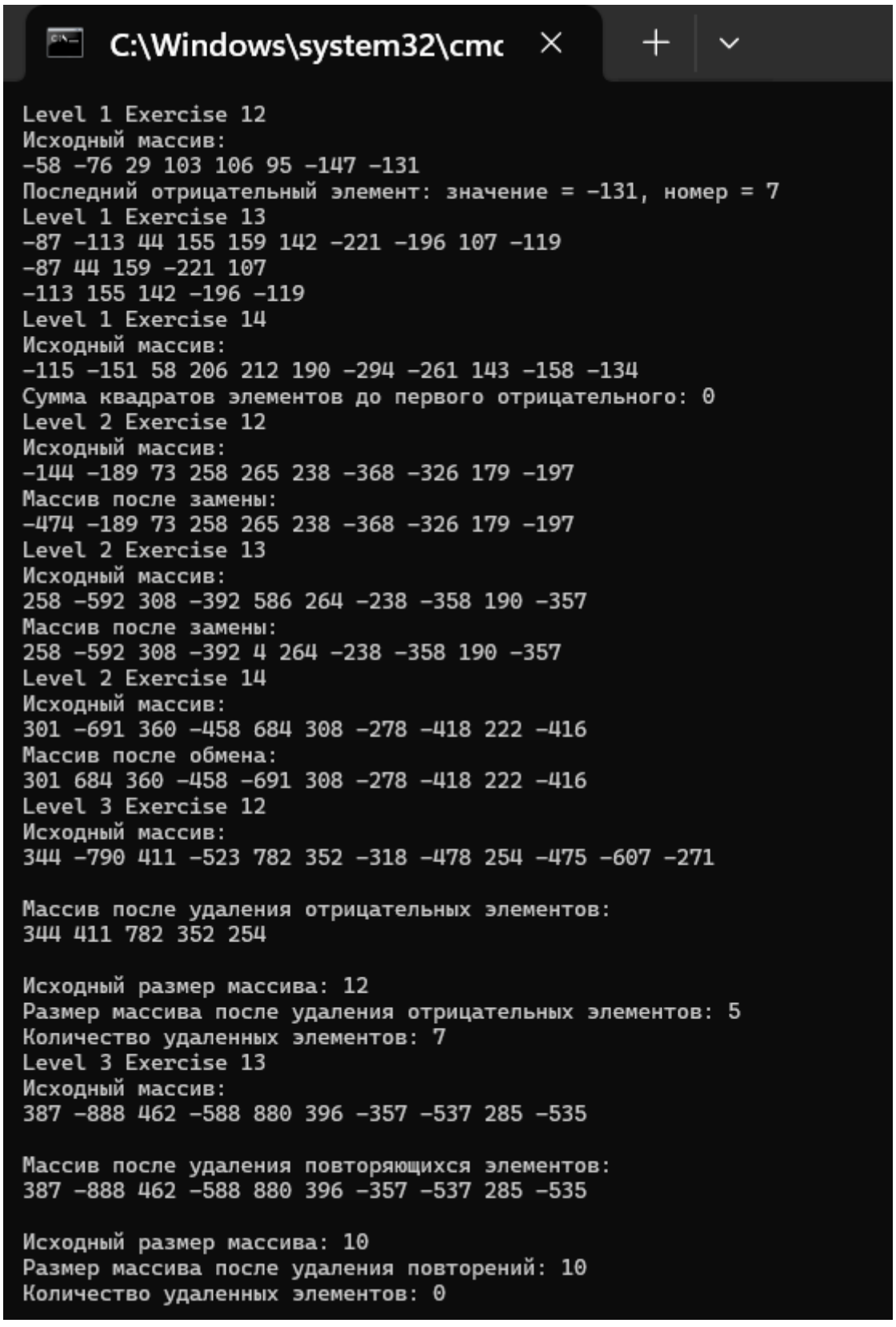
```
{//lvl 2, ex 13
   C.WriteLine("Level 2 Exercise 13");
   int[] a = new int[10]; // based = 10, cause in task no clue what size
   Random rnd = new Random();
   for (int i = 0; i < a.Length; i++)
   {
      a[i] = rnd.Next(-600, 600);
   }
   Console.WriteLine("Исходный массив:");
   Console.WriteLine(string.Join(" ", a));

   int maxEvenIndex = -1;
   int maxEvenValue = int.MinValue;

   for (int i = 0; i < a.Length; i += 2)
   {
      if (a[i] > maxEvenValue)
      {
         maxEvenValue = a[i];
         maxEvenIndex = i;
      }
   }

   if (maxEvenIndex != -1)
   {
      a[maxEvenIndex] = maxEvenIndex;

      Console.WriteLine("Массив после замены:");
      Console.WriteLine(string.Join(" ", a));
   }
   else
   {
      Console.WriteLine("В массиве нет элементов с четными индексами");
   }


}
if (checker == 21)
{//lvl 2, ex 14
   C.WriteLine("Level 2 Exercise 14");
   int[] a = new int[10]; // based = 10, cause in task no clue what size
   Random rnd = new Random();
```

```
    for (int i = 0; i < a.Length; i++)
    {
        a[i] = rnd.Next(-700, 700);
    }
    Console.WriteLine("Исходный массив:");
    Console.WriteLine(string.Join(" ", a));

    int maxIndex = Array.IndexOf(a, a.Max());
    int firstNegativeIndex = Array.FindIndex(a, x => x < 0);

    if (firstNegativeIndex != -1)
    {
        int temp = a[maxIndex];
        a[maxIndex] = a[firstNegativeIndex];
        a[firstNegativeIndex] = temp;

        Console.WriteLine("Массив после обмена:");
        Console.WriteLine(string.Join(" ", a));
    }
    else
    {
        Console.WriteLine("В массиве нет отрицательных элементов");
    }
}



if (checker == 21)
{//lvl 3, ex 12
    C.WriteLine("Level 3 Exercise 12");
    int[] a = new int[12];
    Random rnd = new Random();
    for (int i = 0; i < a.Length; i++)
    {
        a[i] = rnd.Next(-800, 800);
    }
    Console.WriteLine("Исходный массив:");
    Console.WriteLine(string.Join(" ", a));
    //удаление отрицательных
    int[] positiveArray = a.Where(x => x >= 0).ToArray();
```

```
        Console.WriteLine("\nМассив после удаления отрицательных элементов:");
        Console.WriteLine(string.Join(" ", positiveArray));

        Console.WriteLine($"\nИсходный размер массива: {a.Length}");
        Console.WriteLine($"Размер массива после удаления отрицательных элементов: {positiveArray.Length}");
        Console.WriteLine($"Количество удаленных элементов: {a.Length - positiveArray.Length}");
    }


    if (checker == 21)
    {//lvl 3, ex 13
        C.WriteLine("Level 3 Exercise 13");
        int[] a = new int[10];// based = 10, cause in task no clue what size
        Random rnd = new Random();
        for (int i = 0; i < a.Length; i++)
        {
            a[i] = rnd.Next(-900, 900);
        }
        Console.WriteLine("Исходный массив:");
        Console.WriteLine(string.Join(" ", a));

        //Удаление
        int[] uniqueArray = a.Distinct().ToArray();

        Console.WriteLine("\nМассив после удаления повторяющихся элементов:");
        Console.WriteLine(string.Join(" ", uniqueArray));

        Console.WriteLine($"\nИсходный размер массива: {a.Length}");
        Console.WriteLine($"Размер массива после удаления повторений: {uniqueArray.Length}");
        Console.WriteLine($"Количество удаленных элементов: {a.Length - uniqueArray.Length}");

        //Вывод удаленных элементов
        var removedElements = a.GroupBy(x => x)
                                .Where(g => g.Count() > 1)
                                .Select(g => g.Key);
        Console.WriteLine("\nУдаленные элементы (повторения):");
```

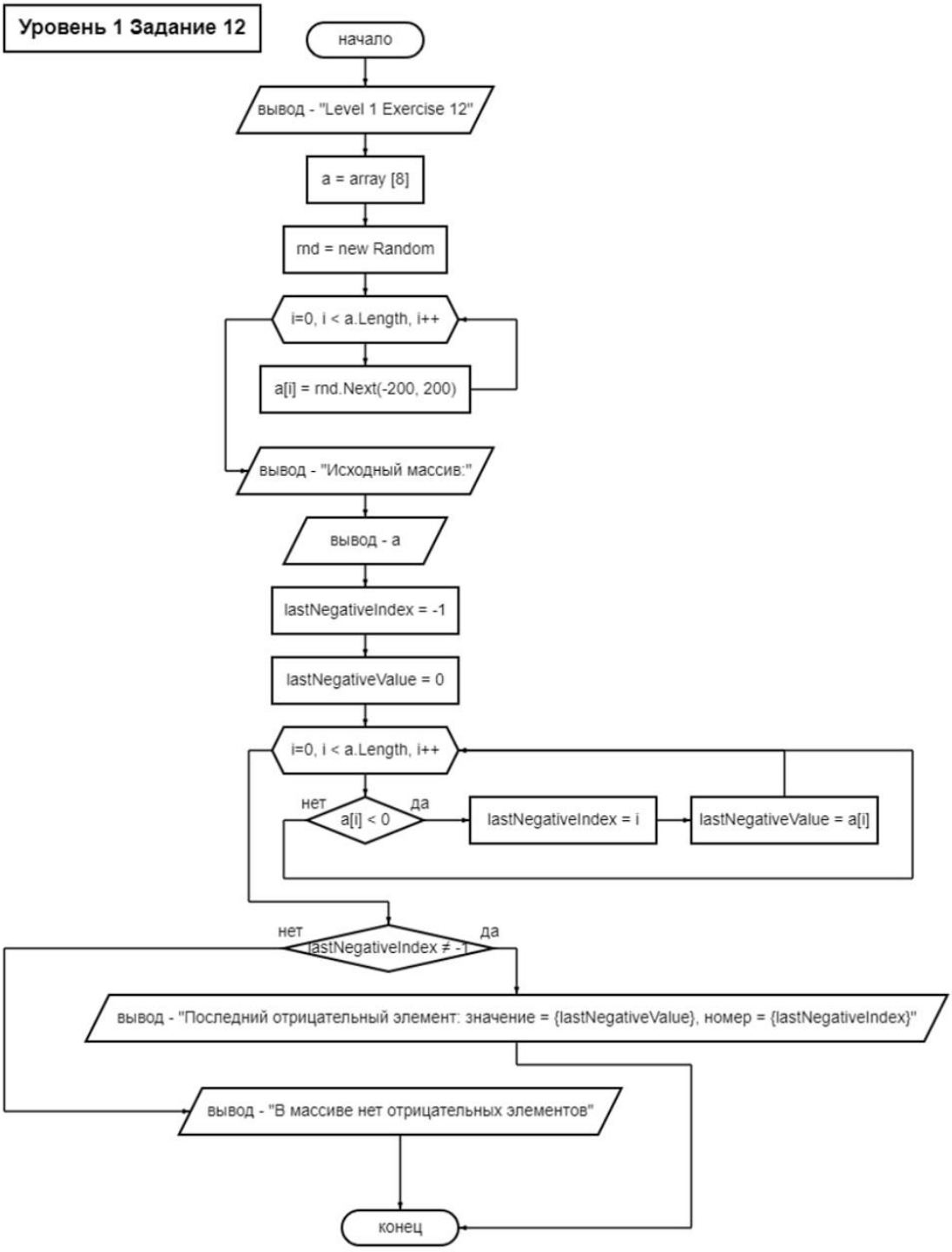
```
            Console.WriteLine(string.Join(" ", removedElements));
        }
    }
}
```

Результат выполнения программы, то есть вывод консоли:

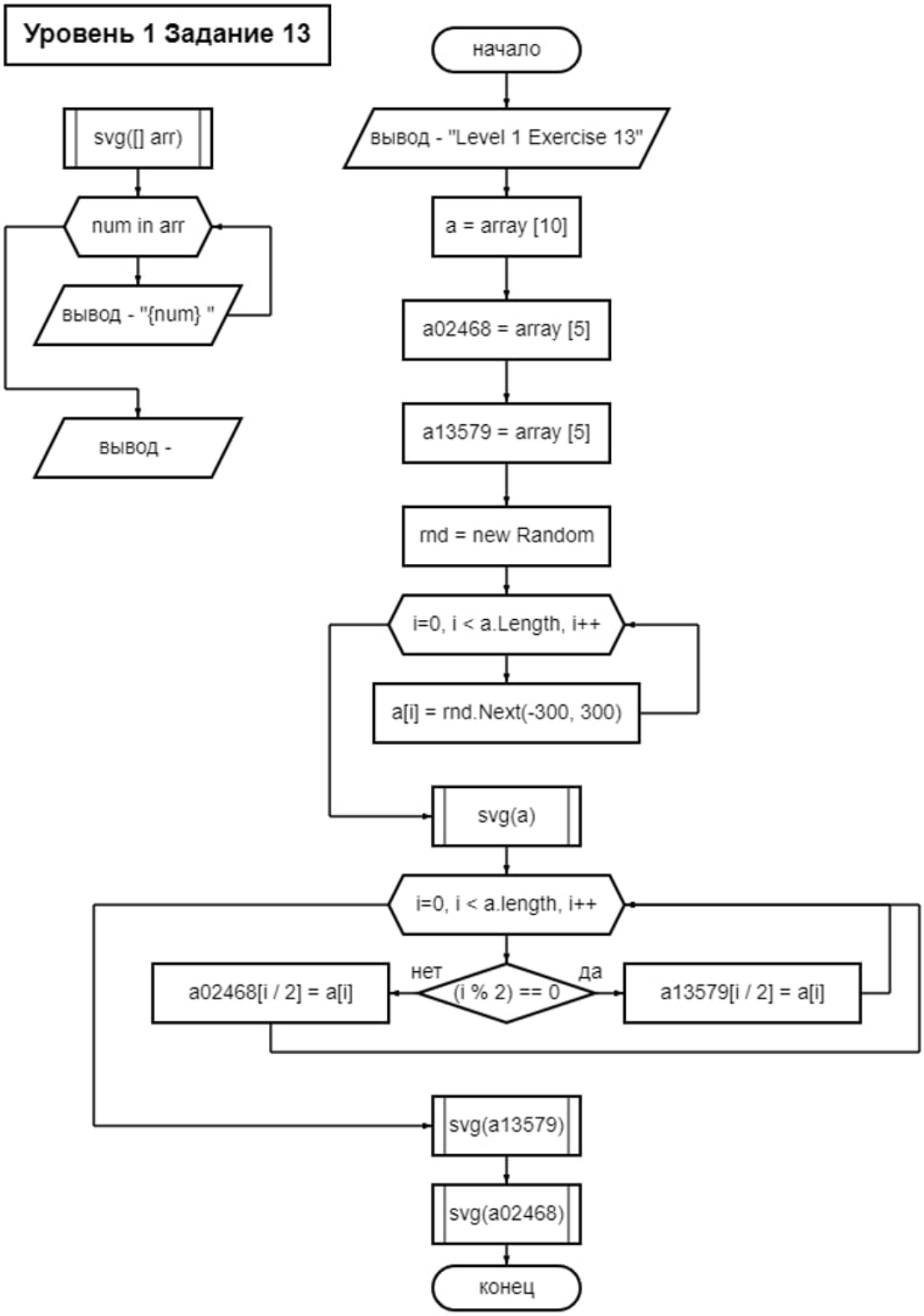


Блок-схемы для каждого задания:

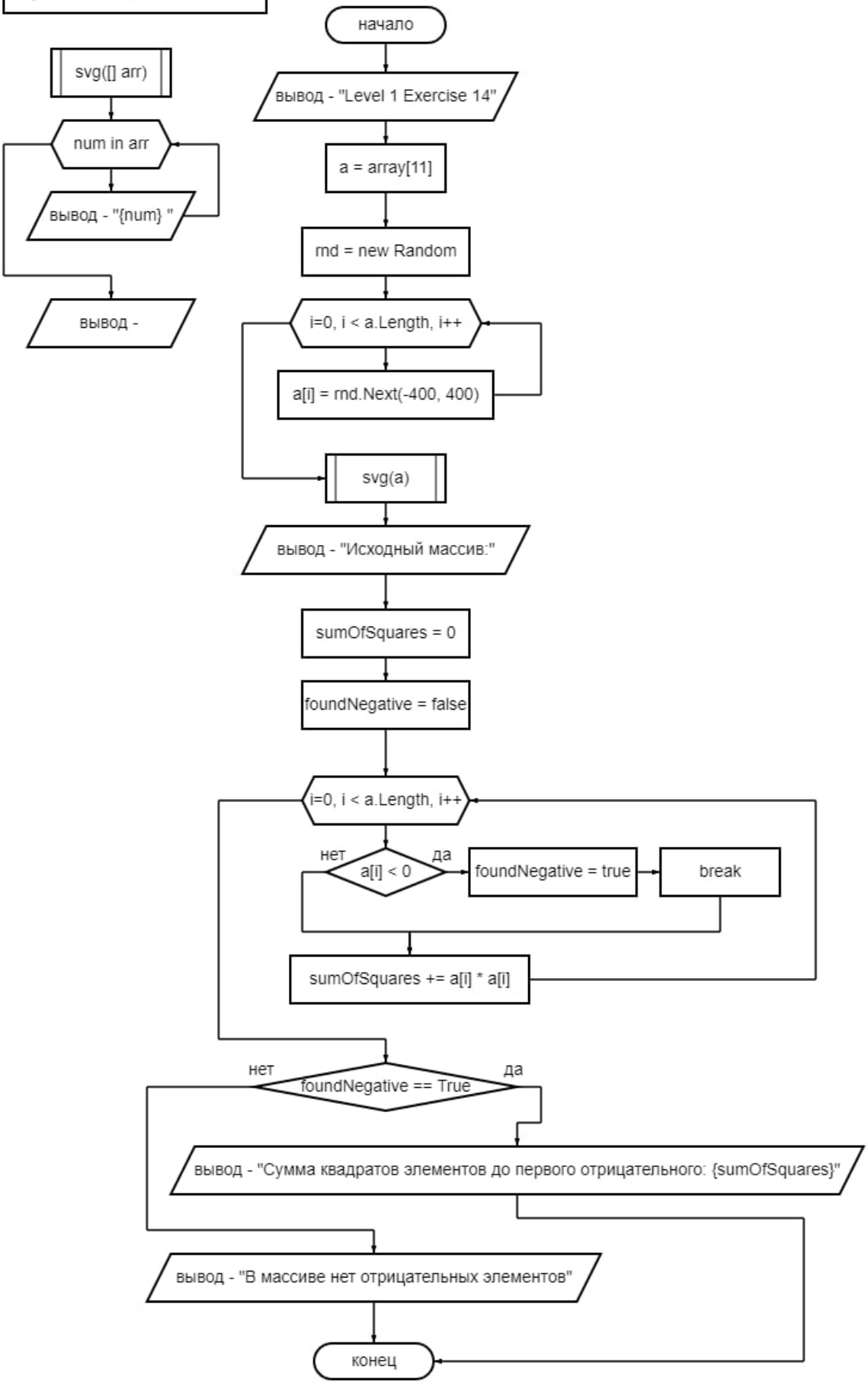
Уровень 1 Задание 12



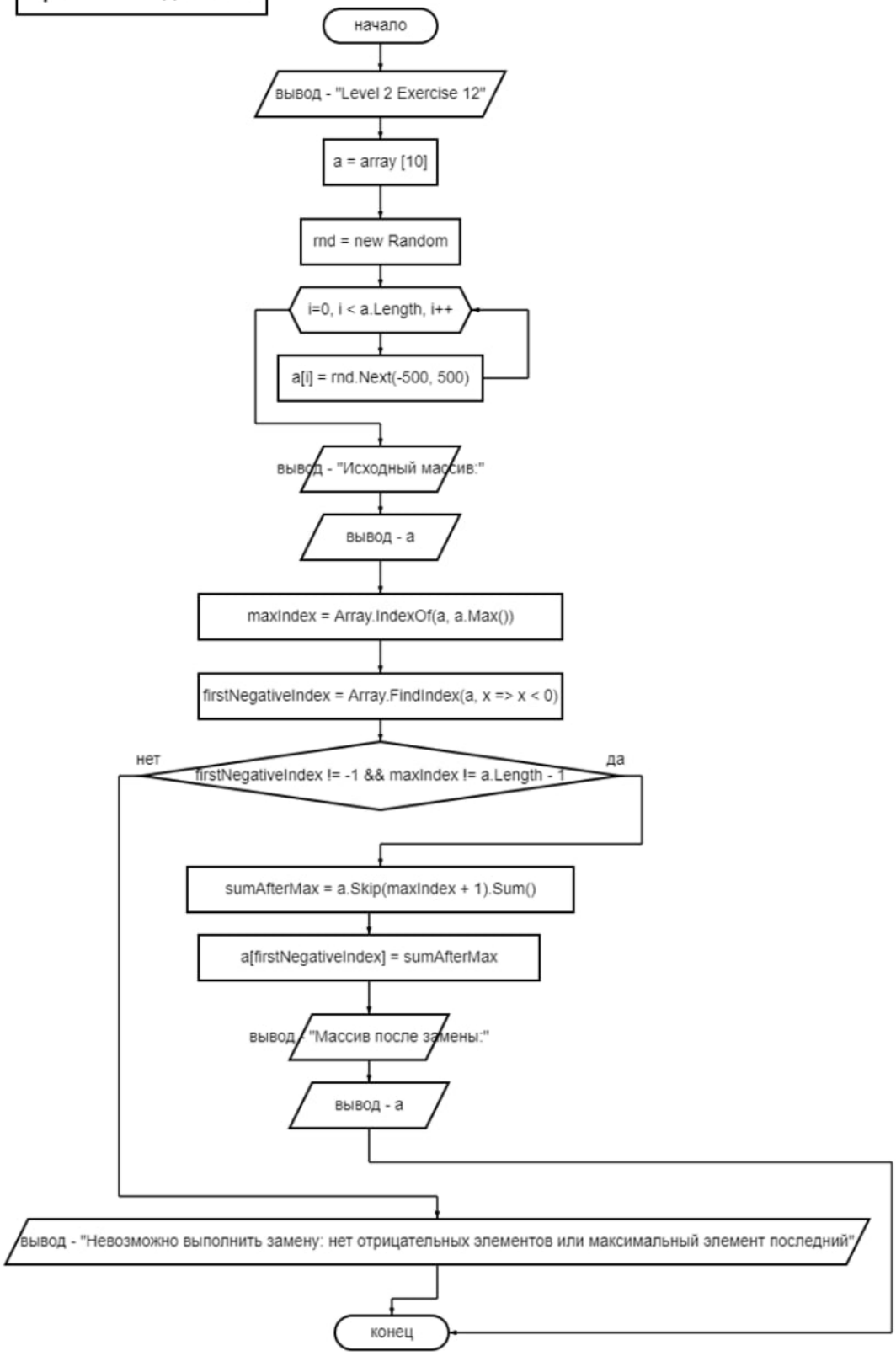
Уровень 1 Задание 13



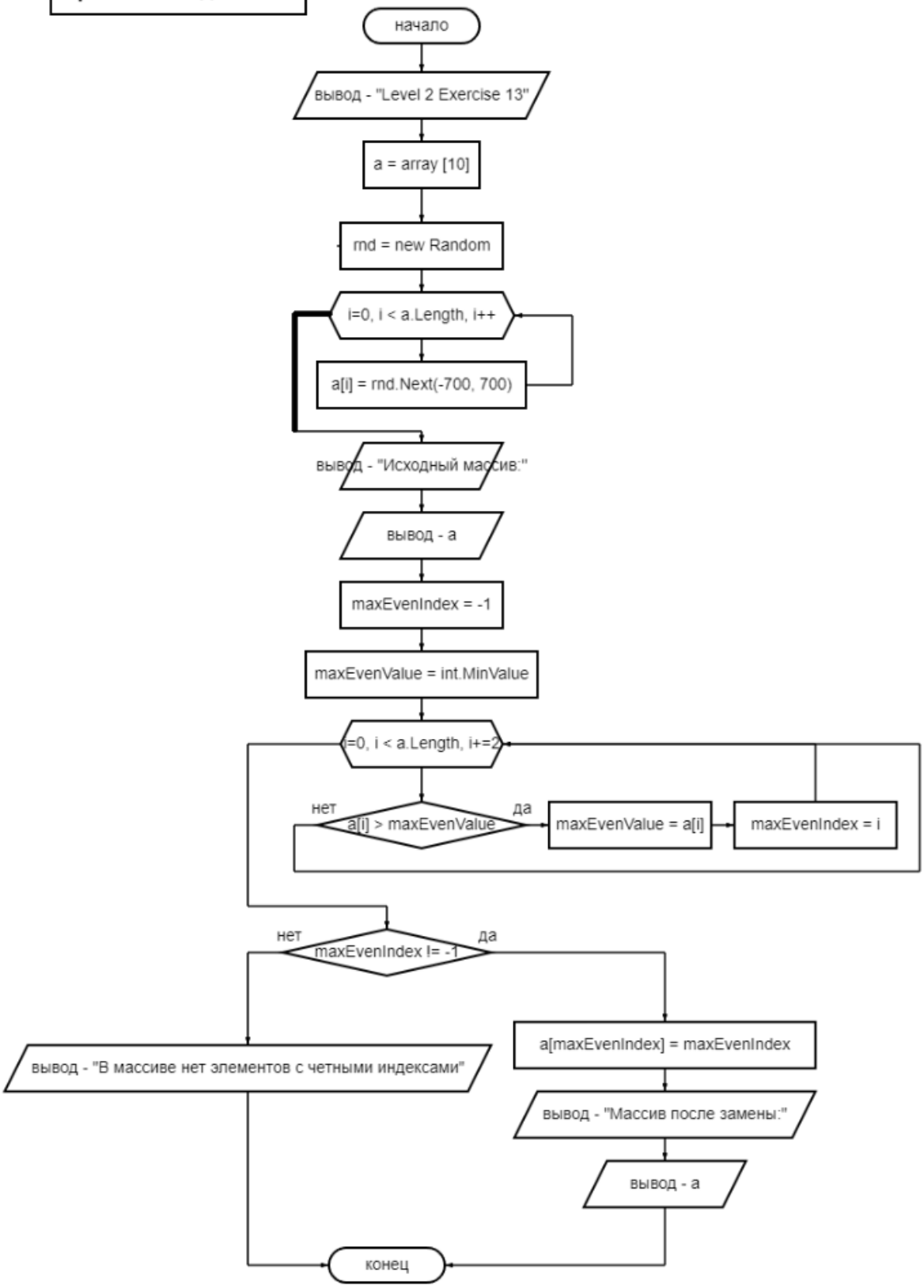
Уровень 1 Задание 14



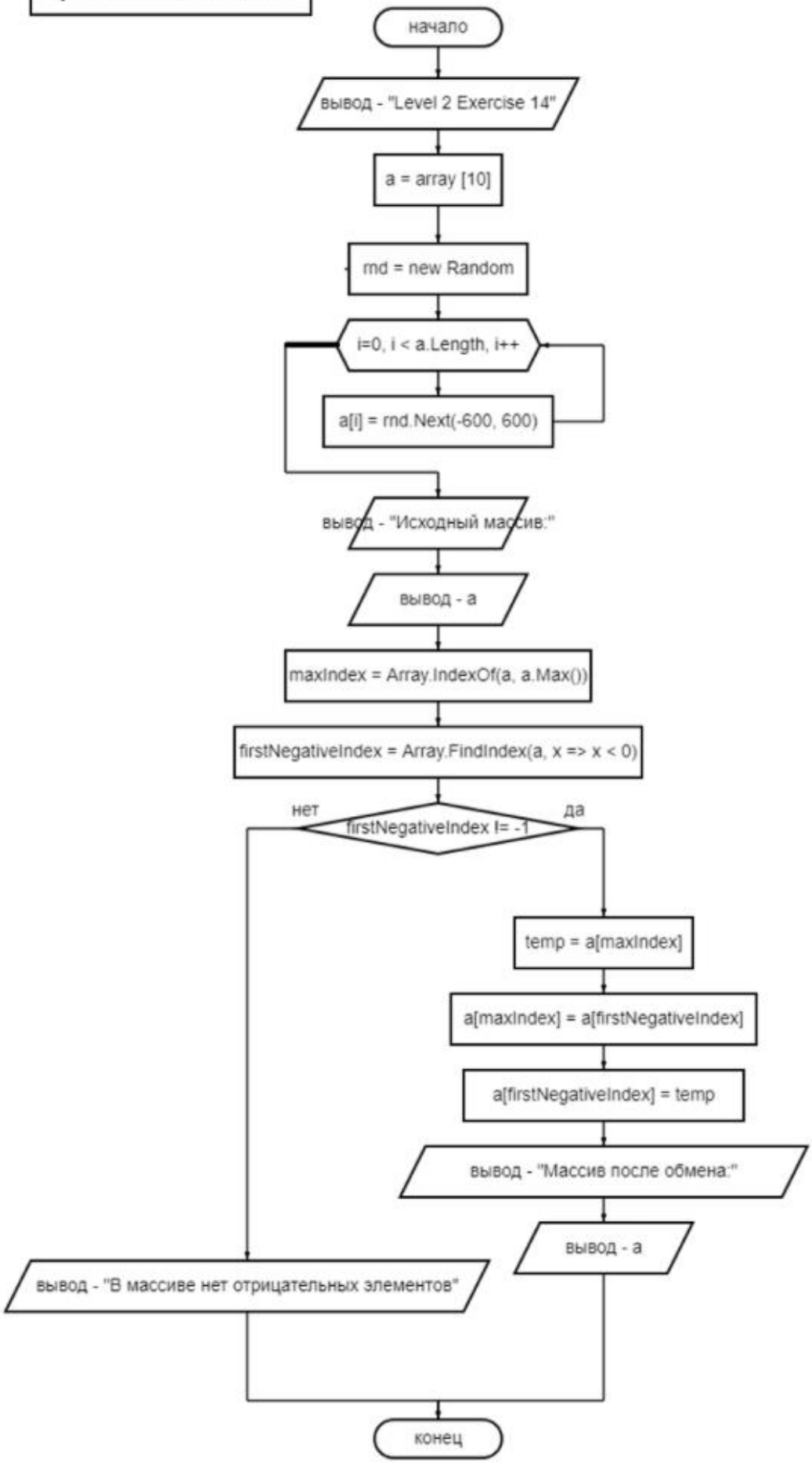
Уровень 2 Задание 12



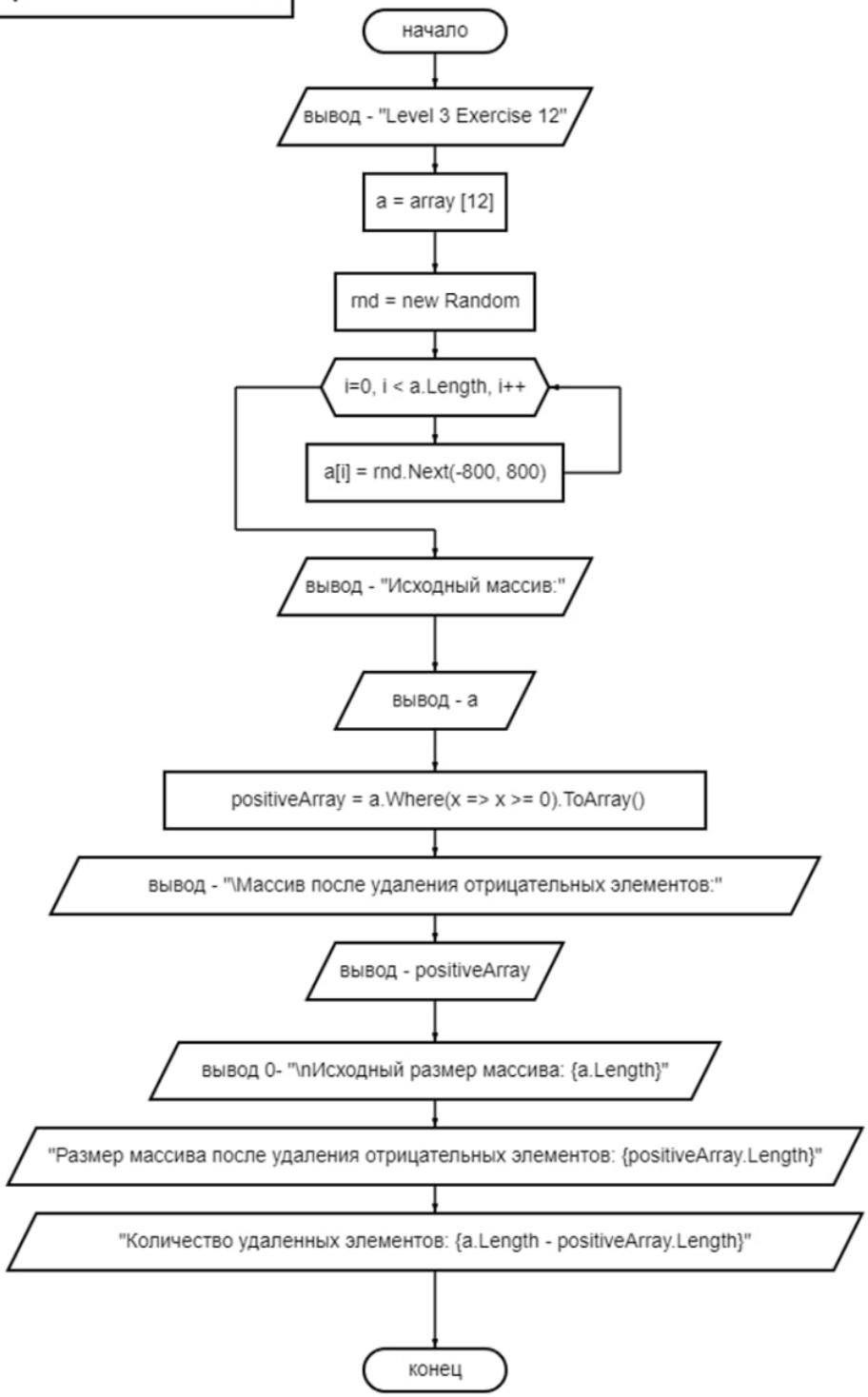
Уровень 2 Задание 13



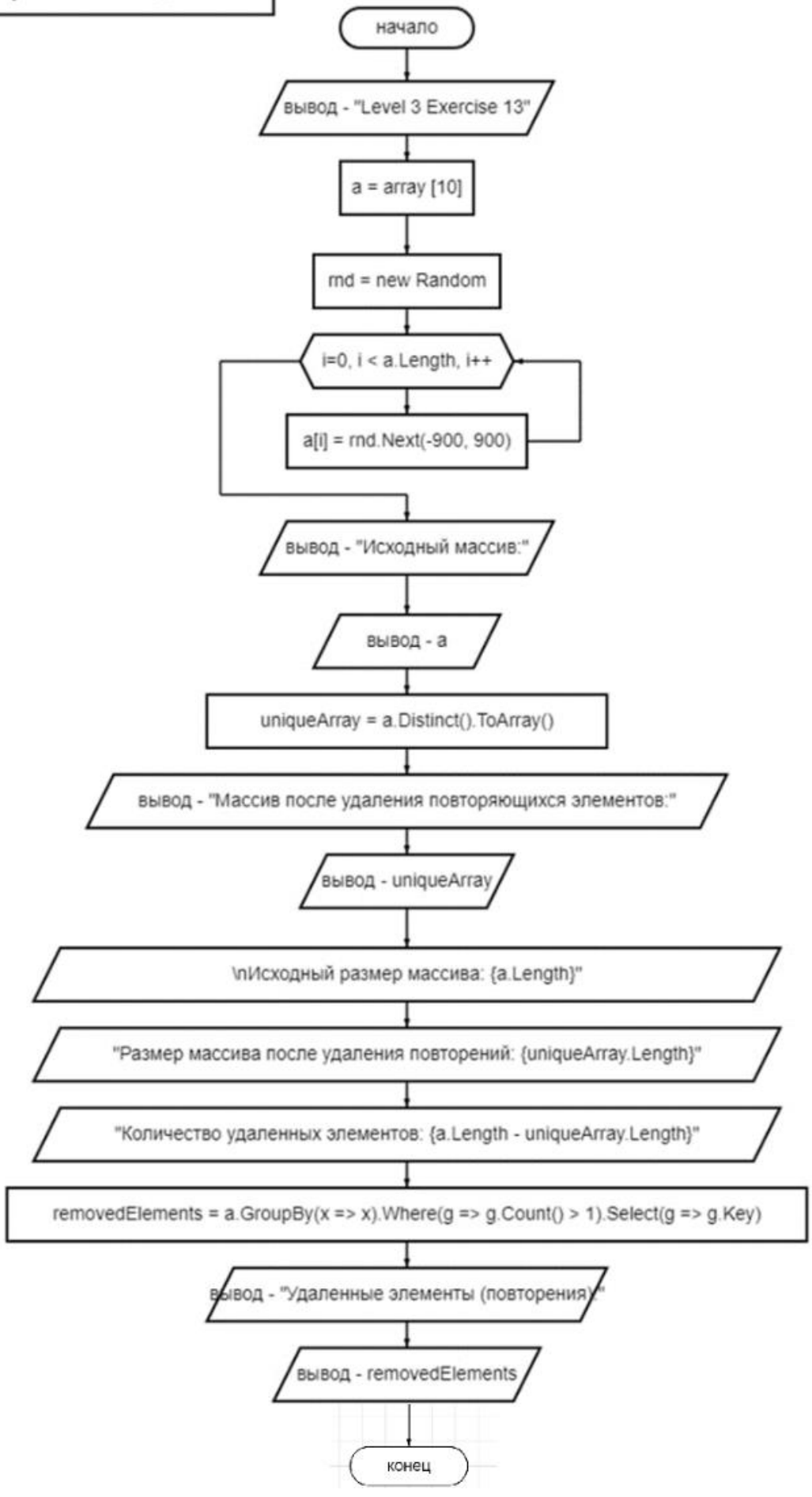
Уровень 2 Задание 14



Уровень 3 Задание 12



Уровень 3 Задание 13



Вывод: в ходе лабораторной работы были успешно реализованы различные алгоритмы обработки одномерных массивов в C#. Программа демонстрирует навыки работы с массивами, включая поиск элементов, удаление отрицательных и повторяющихся значений, замену элементов по условию, и использование LINQ для эффективной обработки данных. Код показывает понимание основных концепций работы с массивами и применение типовых алгоритмов их обработки.