

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное  
бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

---

Кафедра «Информатика»

**Лабораторная работа №9**  
**«Структурированные данные. Программирование**  
**алгоритмов формирования и обработки одномерных**  
**массивов»**

**по дисциплине**  
**«Алгоритмизация и программирование»**

Выполнил: студент гр. БЭИ2202 Кулешов А. С.

Проверил: доц. Воробейчиков Л. А.

Вариант: 17

Москва, 2023 г

## 1) Общее Задание.

123

### 9.1. Общее задание

#### 1) Изучите

вопросы

программирования

алгоритмов

формирования

и

обработки одномерных массивов.

#### 2) Выберите вариант задания из таблицы 1

#### 3) Разработайте проект графического интерфейса пользователя.

#### 4) Решите поставленную задачу формирования и обработки автоматических

одномерных массивов. Для этого разработайте схемы алгоритмов и программные коды следующих функций:

- функции ввода исходных данных, предусмотрев контроль значений исходных данных;
- функции формирования одномерного массива из случайных чисел в заданном диапазоне;
- функции создания нового одномерного массива согласно вашему индивидуальному заданию.

При

необходимости

разбейте

эту

функцию на ряд вспомогательных функций;

- функции вывода одномерного массива в элемент управления

ListBox.

#### 5) Разработайте

программный

код

событийной

процедуры,

которая

вызывает описанные выше функции для решения поставленной задачи.

#### 6) Создайте проект приложения Windows Form, содержащий:

- графический интерфейс;
- файлы с отдельно откомпилированными программными кодами функций, необходимых для решения задачи:
  - функций ввода исходных данных, формирования исходного массива

и вывода результатов;

- функций, решающих поставленную задачу.
- заголовочный файл с прототипами разработанных функций;
- событийную процедуру, которая должна содержать только операторы

вызова пользовательских функций.

7) Подготовьте тесты для контрольного решения задачи, предусмотрев различные значения исходного массива данных для решения. Для проверки работы функции создания нового массива при различных

1

## 2) Индивидуальное задание

17)	Сформировать одномерный массив заданного размера $n$ из случайных целых чисел в заданном диапазоне. Если в нем одинаковое количество положительных и отрицательных элементов, то создать новый массив таким образом, чтобы в первой его половине располагались элементы, стоящие в нечетных позициях, а во второй половине — элементы, стоящие в четных позициях исходного массива.
-----	---

## 3) Формализация и уточнение задания

Из условия задания следует, что его выполнение распадается на следующие этапы, реализуемые соответствующими функциями приложения:

- формирование исходного массива из случайных целых чисел – функция `input`;
- формирование нового массива из простых чисел исходного массива – функция `task`;
- вывод нового массива из простых чисел в заданный список (элемент управления `ListBox`) формы – функция `output`.

Для проверки, является ли элемент исходного массива простым числом, используется вспомогательная функция `simple`.

Вызов функций формирования массивов и вывода нового массива производится в событийной процедуре по нажатию кнопки Решить задачу.

## 4) Схема алгоритма

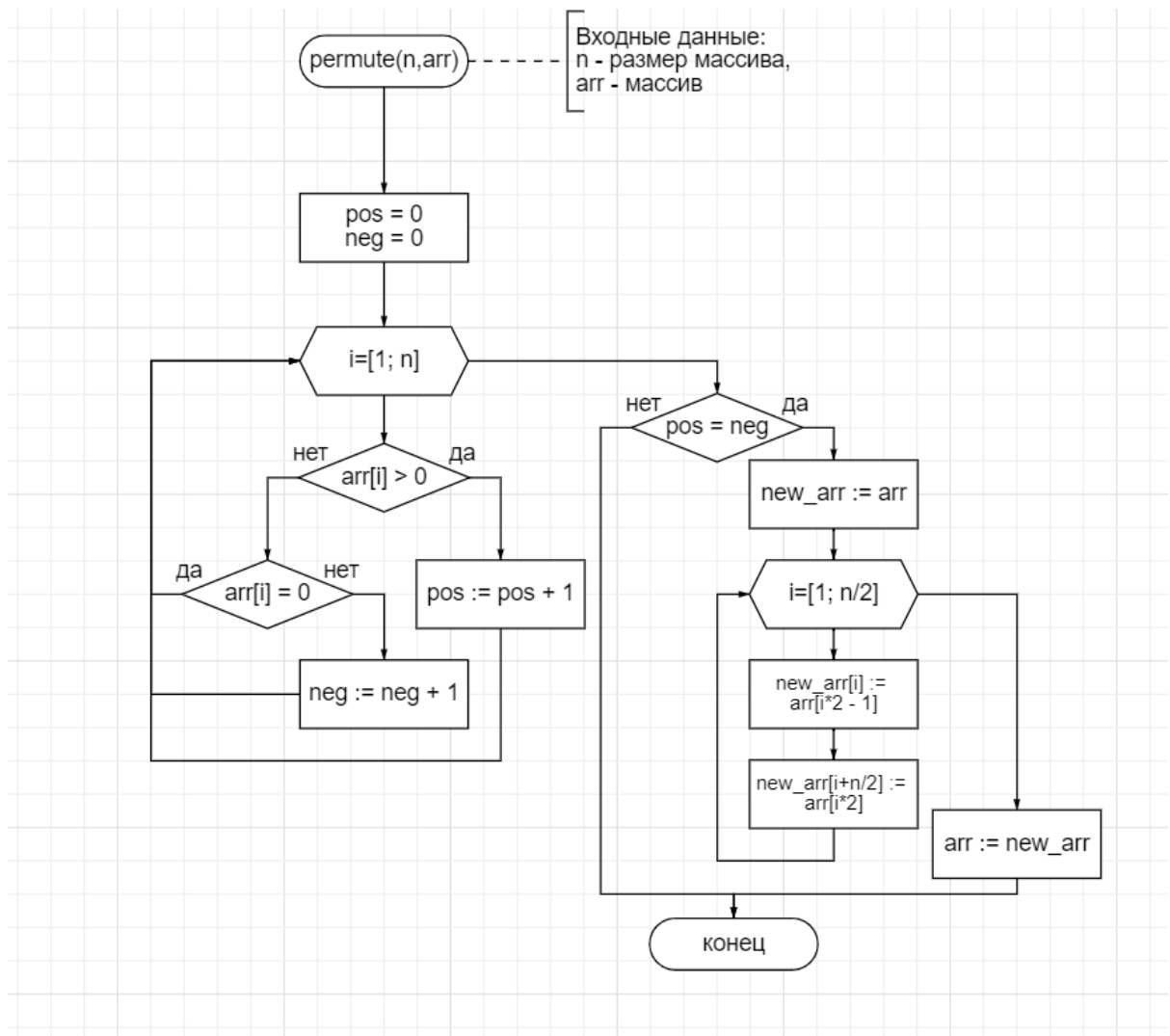


Рисунок 1 – схема алгоритма преобразования массива

5) Теперь создам форму

Одномерные Статические Массивы

Размер исходного массива

Диапазон случайных чисел a  b

Исходный массив listBox1

полученный массив listBox2

Решение

Выход

Рисунок 2 – полученная форма

Теперь реализую код для кнопок

Ссылка: 1

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    listBox2.Items.Clear();

    int n = int.Parse(textBox1.Text);
    int a = int.Parse(textBox2.Text);
    int b = int.Parse(textBox3.Text);
    int[] arr = new int[n];
    Random rnd = new Random();
    for (int i = 0; i < n; ++i)
    {
        arr[i] = rnd.Next() % (b-a+1) + a;
        listBox1.Items.Insert(i, arr[i]);
        Console.WriteLine(arr[i]);
    }
    arr = permute(n, arr);

    for (int i = 0; i < n; ++i)
    {
        listBox2.Items.Insert(i, arr[i]);
    }
}
```

Ссылка: 1

```
private void button2_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Рисунок 3 – код для кнопок

Ссылка: 1

```
int[] permute(int n, int[] arr)
{
    int pos = 0;
    int neg = 0;

    for (int i = 0; i < n; ++i)
    {
        if (arr[i] > 0)
        {
            pos++; continue;
        }

        if (arr[i] < 0)
        {
            neg++;
        }
    }

    if (neg != pos) return arr;

    int[] new_arr = new int[n];

    for(int i = 0; i < n/2; ++i)
    {
        new_arr[i] = arr[i * 2];
        new_arr[n / 2 + i] = arr[i * 2 + 1];
    }
    return new_arr;
}
```

Рисунок 4 – код функции permute

Наконец, проверю работоспособность алгоритма на тестовых данных

Размер исходного массива	<input type="text" value="6"/>		
	a	b	
Диапазон случайных чисел	<input type="text" value="-1"/>	<input type="text" value="1"/>	<input type="button" value="Решение"/>
Исходный массив	полученный массив		
<div>0 0 1 0 1 0</div>	<div>0 0 1 0 1 0</div>		<input type="button" value="Выход"/>

Рисунок 5 – Первый набор тестовых данных

Одномерные Статические Массивы

Размер исходного массива

Диапазон случайных чисел      a      b  
        

Исходный массив      полученный массив

-1	-1
-1	0
0	0
1	-1
0	1
1	1

Рисунок 6 – Второй набор тестовых данных

Размер исходного массива

Диапазон случайных чисел      a      b  
        

Исходный массив      полученный массив

0	0
-1	-1
1	1
0	0
-1	-1
0	0

Рисунок 7 – Третий набор тестовых данных



Одномерные Статические Массивы

Размер исходного массива

Диапазон случайных чисел a  b

Исходный массив

1
-1
-1
1
-1
1

полученный массив

1
-1
-1
1
-1
1

Рисунок 8 – Четвёртый набор тестовых данных

Как можно заметить в первом и третьем наборе тестовых данных количество отрицательных и положительных элементов не совпадает, поэтому массив не меняется.

Во втором и четвёртых примерах количество отрицательных и положительных элементов совпадает, поэтому массив изменяется.

Программа работает корректно