

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра Вычислительной техники**

**ОТЧЕТ**  
**по лабораторной работе № 2**  
**по дисциплине «Программирование»**  
**ТЕМА: «ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ»**

Студент гр. 3311

\_\_\_\_\_

Шарпинский Д. А.

Преподаватель

\_\_\_\_\_

Хахаев И. А.

Санкт-Петербург

2023

### **Цель работы.**

Целью работы является изучение одномерных массивов и получение практических навыков в их обработке.

### **Задание (вариант 14)**

Определить, есть ли в массиве хотя бы одна тройка соседних чисел, в которой средний элемент больше своих «соседей», т. е. предшествующего и последующего. В случае положительного ответа определить номера элементов первой из таких троек.

### **Постановка задачи и описание решения**

Предоставлена задача на определение наличия хотя бы одной тройки соседних чисел в массиве, в которой средний элемент больше своих соседей, то есть предшествующего и последующего. Необходимо также определить номера элементов первой из таких троек, игнорируя оставшиеся, если такие были. Для выполнения задачи необходимо написать программу.

1. Инициализируются переменные `arr[10]` - массив размером 10 элементов для хранения введенных чисел, `i`, `flag`.
2. С помощью цикла `for`, пользователь вводит 10 чисел, по одному за раз. Каждое число сохраняется в соответствующем элементе массива.
3. Переменной `flag` передается значение 1. Этот флаг используется для определения, была ли найдена тройка чисел, удовлетворяющая условию задачи.
4. С помощью еще одного цикла `for`, проходящего по массиву, проверяется каждая тройка соседних чисел начиная с первого элемента и до предпоследнего (для рассмотрения троек не должно возникнуть ошибки выхода за пределы массива, а потому мы не берем крайним значением 10, вместо этого цикл будет идти только до 8, чтобы при `i == 7` элемент,

стоящий на 2 индекса впереди, как раз оказался последним). Для каждой тройки выполняется следующая проверка:

5. Если флаг равен 1 (то есть еще не найдена подходящая тройка), и текущий элемент меньше следующего элемента ( $arr[i] < arr[i+1]$ ) и следующий элемент больше последующего элемента ( $arr[i+1] > arr[i+2]$ ), то флаг устанавливается в 0, и выводятся номера элементов, составляющих тройку.
6. Если после прохода цикла флаг все еще равен 1, это означает, что в массиве не было найдено подходящей тройки чисел, и выводится соответствующее сообщение.

### Описание переменных

№	Имя переменной	Тип	Назначение
1	arr	int	Хранение последовательности чисел
2	i	int	Переменная в цикле
3	flag	int	Отслеживает наличие подходящей тройки чисел.

### Контрольные примеры

#### Пример 1:

arr = [1, 2, 3, 4, 6, 5, 7, 8, 9, 10]

answer: The numbers of the required triple have the following numbers: 4 5 6

#### Пример 2:

arr = [1, 1, 1, 1, 1, 1, 1, 1, 1, 2]

answer: The array does not contain the required triple of numbers

## Примеры выполнения программы

```
Enter the number of subsequence (10 numbers left): 1
Enter the number of subsequence (9 numbers left): 2
Enter the number of subsequence (8 numbers left): 3
Enter the number of subsequence (7 numbers left): 4
Enter the number of subsequence (6 numbers left): 6
Enter the number of subsequence (5 numbers left): 5
Enter the number of subsequence (4 numbers left): 7
Enter the number of subsequence (3 numbers left): 8
Enter the number of subsequence (2 numbers left): 9
Enter the number of subsequence (1 numbers left): 10

Your array is: [1, 2, 3, 4, 6, 5, 7, 8, 9, 10]

The numbers of the required triple have the following numbers: 4 5 6

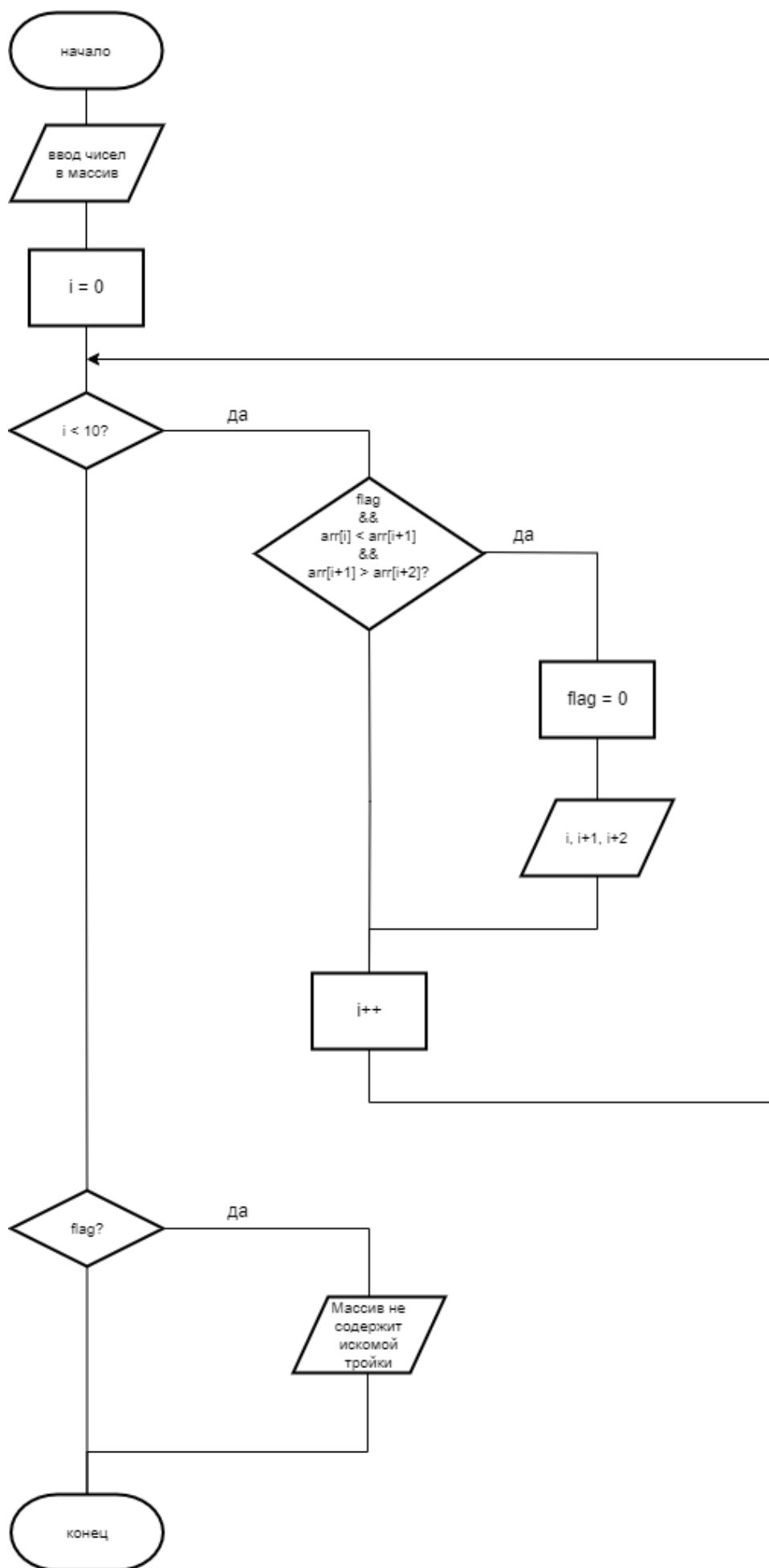
Process returned 0 (0x0)   execution time : 9.833 s
Press any key to continue.
```

```
Enter the number of subsequence (10 numbers left): 1
Enter the number of subsequence (9 numbers left): 1
Enter the number of subsequence (8 numbers left): 1
Enter the number of subsequence (7 numbers left): 1
Enter the number of subsequence (6 numbers left): 1
Enter the number of subsequence (5 numbers left): 1
Enter the number of subsequence (4 numbers left): 1
Enter the number of subsequence (3 numbers left): 1
Enter the number of subsequence (2 numbers left): 1
Enter the number of subsequence (1 numbers left): 2

Your array is: [1, 1, 1, 1, 1, 1, 1, 1, 1, 2]

The array does not contain the required triple of numbers

Process returned 0 (0x0)   execution time : 4.959 s
Press any key to continue.
```



```

1  #include <stdio.h>
2
3  int main() {
4
5      int arr[10], i, flag;
6
7      for (i = 0; i < 10; i++) {
8          printf("Enter the number of subsequence (%d numbers left): ", 10 - i);
9          scanf("%d", &arr[i]);
10     }
11
12     printf("\nYour array is: [");
13     for (i = 0; i < 9; i++) {
14         printf("%d, ", arr[i]);
15     }
16     printf("%d]\n\n", arr[9]);
17
18     flag = 1;
19     for (i = 0; i < 8; i++) {
20         if (flag && arr[i] < arr[i+1] && arr[i+1] > arr[i+2]) {
21             flag = 0;
22             printf("The numbers of the required triple have the following numbers: %d %d %d\n", i+1, i+2, i+3);
23         }
24     }
25
26     if (flag) {
27         printf("The array does not contain the required triple of numbers\n");
28     }
29
30     return 0;
31 }
32

```

### **Выводы.**

В результате выполнения работы изучены одномерные массивы и получены практические навыки в их обработке.