

Министерство науки и образования РФ  
Федеральное государственное автономное образовательное  
учреждение высшего профессионального образования  
«Санкт-Петербургский государственный электротехнический  
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»  
(СПбГЭТУ «ЛЭТИ»)  
Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчет по лабораторной работе №2  
на тему:  
“Одномерные массивы в языке C/C++”  
по дисциплине “Программирование”

Выполнил: Табаков А.В.  
Группа 4306  
Принял: к.т.н., доцент Сискович Т.И.

Санкт-Петербург  
2014 г.

**Цель**

Получить практические навыки обработки одномерных массивов на языке «C/C++».

### Задание

Разработать алгоритм и написать программу, которая сортирует массив по убыванию и определяет, присутствует ли в массиве отрицательный элемент. Программа выполняется многократно, по желанию пользователя.

### Уточнение задания

Вначале программа должна предложить пользователю задать размер массива. При этом программа должна проверить введённое пользователем значение. Размер массива должен быть больше 0, но меньше константы n. После контрольного вывода, программа выводит обработанный массив и сообщение о присутствии отрицательного числа. Далее программа должна предложить пользователю заново ввести элементы массива и/или размера массива с помощью переменных, значения которых вводятся с клавиатуры и соответствующих организации циклов.

### Контрольные примеры

Контрольные примеры представлены в Таблице 1.

Таблица 1. Контрольные примеры

№	Исходные данные		Результат	
	size	mas[size]	Сортировка по убыванию	Отрицательное число
1	0	-	Кол-во элементов, целые положительные числа, от 1 до n! Введите кол-во повторно!	
2	5	1 8 0 3 4	8 4 3 1 0	Отрицательное число отсутствует
3	6	9 4 -5 3 4 2	9 4 4 3 2 -5	Отрицательное число присутствует
4	3	2 3 0	3 2 0	Отрицательное число отсутствует
5	4	0 3 9 2	9 3 2 0	Отрицательное число отсутствует

## Описание переменных

Описание переменных представлено в Таблице 2.

Таблица 2. Описание переменных

Имя переменной	Тип	Назначение
i	int	Вспомогательная переменная для организации цикла
j	int	Вспомогательная переменная для организации цикла
size	int	Размер массива
f	int	Булева переменная
buf	int	Вспомогательная переменная для сортировки
mas	int	Массив
Q	char	Вспомогательная переменная для организации цикла
n	const int	Константа для определения размера массива

## Описание алгоритма

Шаг №1 Вывод сообщения пользователю с просьбой ввести размер массива.

Шаг №2 Ввод пользователем размера массива.

Шаг №3 Если размер массива меньше 0 или больше n, то вывод сообщения об ошибке и переход к шагу №2, иначе переход к шагу №4.

Шаг №4 Вывод сообщения пользователю с просьбой ввести элементы массива.

Шаг №5 Ввод пользователем всех элементов массива через пробел.

Шаг №6 Проверка на отрицательные числа, если они есть f=1.

Шаг №7 Контрольный вывод элементов массива.

Шаг №8 Сортировка массива по убыванию.

Шаг №9 Вывод отсортированного массива.

Шаг №10 Вывод сообщения об отриц. элементе массива.

Шаг №11 Вывод сообщения: "Повторить ввод чисел в массив при заданном кол-ве(size)? Для повтора введите 'Y'".

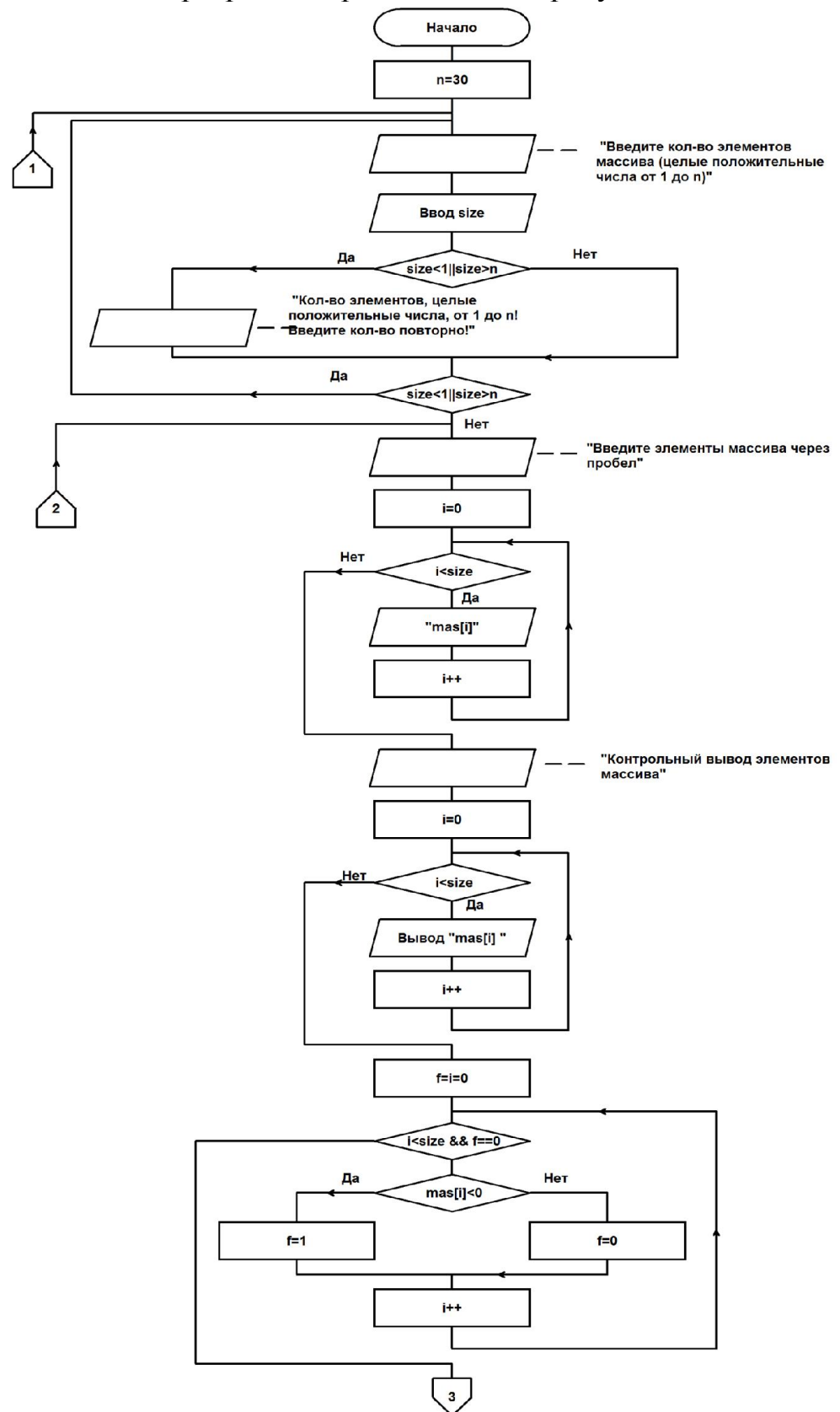
Шаг №12 Если на вопрос пользователь отвечает 'Y' или 'y', то осуществляется переход к шагу № 4.

Шаг №13 Вывод сообщения: «Повторить программу заново? Для повтора введите 'Y'».

Шаг №14 Если на вопрос пользователь отвечает 'Y' или 'y', то осуществляется переход к шагу № 1, иначе программа завершается.

## Блок-схема программы

Блок-схема программы представлена на рисунке 2



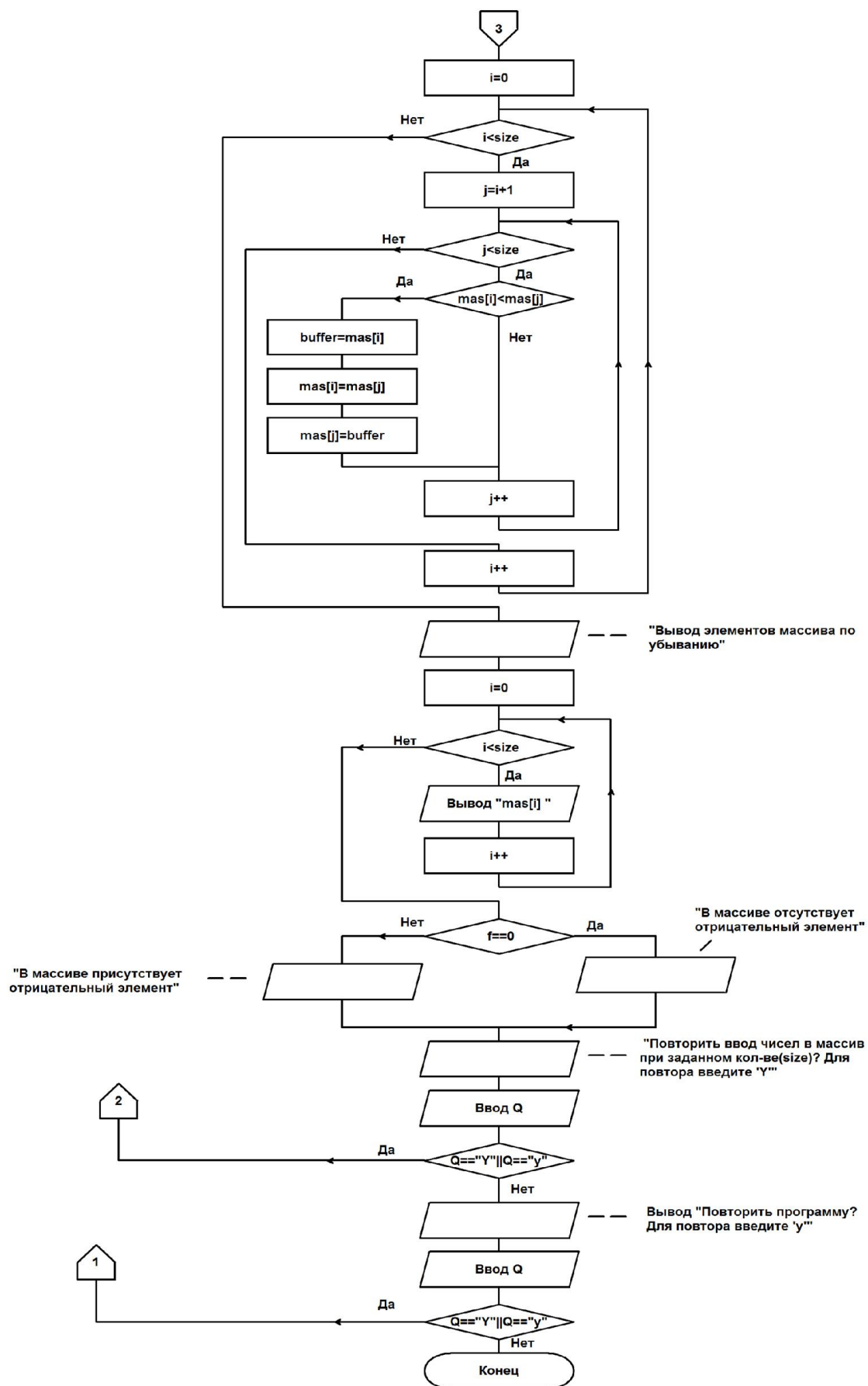


Рис.2. Блок-схема

## Текст программы с комментариями

```
#include<stdio.h>
#include<locale.h>
int main()
{
    setlocale(LC_ALL, "RUS");
    const int n=30;
    int mas[n], i, j, size, buffer, f;
    char Q;
    do
    {
        printf("Введите кол-во элементов массива (целые положительные числа от 1 до n)\n");
        do
        {
            scanf("%d", &size);
            if (size<1||size>n)
                printf("Кол-во элементов, целые положительные числа, от 1 до n! Введите кол-во повторно!\n");
        }
        while (size<1||size>n);
        do
        {
            printf("Введите элементы массива через пробел\n");
            for (i = 0; i < size; scanf("%d", &mas[i]), i++);
            for (f = i = 0; i < size && f==0 ; f=(mas[i]<0)?1:0, i++);
            printf("Контрольный вывод элементов массива\n");
            for (i = 0; i < size; printf("%d ", mas[i]), i++);
            for (i = 0; i < size; i++)
                for (j = i + 1; j < size + 1; j++)
                    if (mas[i] < mas[j])
                    {
                        buffer = mas[i];
                        mas[i] = mas[j];
                        mas[j] = buffer;
                    }
            printf("\nВывод элементов массива по убыванию\n");
            for (i = 0; i < size; printf("%d ", mas[i]), i++);
            if (f!=0)
                printf("\nВ массиве присутствует отрицательный элемент")
            else
                printf("\nВ массиве отсутствует отрицательный элемент");
            printf("\nПовторить ввод чисел в массив при заданном кол-ве(%d)? Для повтора введите 'Y'\n", size);
            fflush(stdin);
            scanf("%c", &Q);
        }
        while (Q == 'Y' || Q == 'y');
        printf("\nПовторить программу заново? Для повтора введите 'Y'\n");
        fflush(stdin);
        scanf("%c", &Q);
    }
    while (Q == 'Y' || Q == 'y');
```

```
return 0;  
}
```

### **Результаты решения задачи**

При выполнении программы были получены результаты, совпадающие со значениями, приведенными в Таблице 1. Ошибок не обнаружено.

### **Вывод**

При выполнении лабораторной работы были получены практические навыки обработки одномерных массивов на языке программирования «C/C++».