

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Программирование»**  
**Тема: Управляющие конструкции языка СИ**

Студент гр. 9382

\_\_\_\_\_

Савельев И.С.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2019

### **Цель работы.**

Изучить управляющие конструкции в языке C. Научиться работать с разными типами данных, массивами, циклами и условными операторами. Освоить использование функций.

### **Задание**

Вариант №5. Реализуйте программу, на вход которой подается одно из значений 0, 1, 2, 3 и массив целых чисел размера не больше 100. Числа разделены пробелами. Строка заканчивается символом перевода строки.

В зависимости от значения, функция должна выводить следующее:

0 : максимальное по модулю число в массиве. (abs\_max)

1 : минимальное по модулю число в массиве. (abs\_min)

2 : разницу между максимальным по модулю и минимальным по модулю элементом. (diff)

3 : сумму элементов массива, расположенных после максимального по модулю элемента (включая этот элемент). (sum)

Иначе необходимо вывести строку "Данные некорректны".

### **Основные теоретические положения.**

Выполнение работы производилось при помощи текстового редактора Codeblocks. Были использованы библиотеки `<stdio.h>` и `<stdlib.h>`. Правильность выполнения программы оценивалась при помощи проверяющей системы stepik.

### **Выполнение работы.**

1) В начале работы были подключены стандартные библиотеки языка C: `<stdio.h>` и `<stdlib.h>`. Далее была создана функция `main()`.

2) В функции `main()` были объявлены: массив из 100 элементов, счетчик `i` для заполнения массива, переменная `flag`, служащая для определения конца считывания элементов массива. Переменная типа `int menu`, которая

регистрировала введенное значение, впоследствии использованное в логике программы. А также целочисленная переменная `size`, отвечающая за количество введенных с клавиатуры элементов массива. Далее посредством цикла `while()` было реализовано присвоение значений элементам массива.

3) Следующим шагом было объявление и описание функции типа `int abs_max()`, которая принимает в качестве аргументов некоторый массив и количество обрабатываемых элементов. В теле функции с помощью цикла `for` был реализован поиск максимального по модулю элемента в массиве, что и является возвращаемым значением данной функции.

4) Далее была описана функция `abs_min` аналогичная по строению с функцией `abs_max`. Однако в данном случае, возвращаемым значением этой функцией было значение минимального по модулю элемента в массиве. Реализован поиск этого значения с помощью цикла `for` и сравнением элементов с переменной `min`, объявленной в теле функции.

5) Далее была описана функция `diff()`, возвращающая разность между максимальным и минимальным по модулю элементами массива. В теле функции использовались уже описанные функции `abs_max` и `abs_min`.

6) Следующим шагом было описание функции `sum()`, которая складывает значения элементов массива, имеющие индекс больший или равный индексу максимального по модулю элемента. В теле функции было реализовано суммирование таких элементов с помощью цикла `for` и использования функции `abs_max()`.

7) В функции `main()` затем с помощью конструкции `switch()` был реализован вывод в зависимости от значения `val`, введенного пользователем в самом начале выполнения программы.

Исходный код см. в приложении А.

## **Тестирование**

Результаты тестирования представлены в табл. 1.

Таблица 1 — результаты тестирования.

№ п/п	Входные данные	Выходные данные	Комментарий
0	42 6 4 0 -34 5 8 0 -1	42	Максимальный по модулю элемент.
1	1 2 74 2 4 -33 -21 3 0 3	0	Минимальный по модулю элемент.
2	2 4 9 1 33 -11 6 76 2 -3 4	75	Разница между максимальным и минимальным по модулю элементами.
3	3 6 19 0 -1 4 8 12 60 -4 5	61	Сумма элементов, стоящих после максимального по модулю элемента.
4	10 8 11 -3 9 33 50 82 0	Данные некорректны	Первое введенное число не равно ни 0, ни 1, ни 2, ни 3.

### Выводы.

Были изучены управляющие конструкции языка и стандартные библиотеки языка C. Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. Для обработки команд пользователя использовался оператор switch, ввод осуществлялся через цикл while().

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ.

```
#include <stdio.h>
#include <stdlib.h>

int abs_max(int array[], int size)
{
    int max = array[0];
    for (int i = 0; i < size; i++)
    {
        if (abs(array[i]) > abs(max))
        {
            max = array[i];
        }
    }
    return max;
}

int abs_min(int array[], int size)
{
    int min = array[0];
    for (int i = 0; i < size; i++)
    {
        if (abs(array[i]) < abs(min))
        {
            min = array[i];
        }
    }
    return min;
}

int diff(int array[], int size)
{
    return abs_max(array, size) - abs_min(array, size);
}

int sum(int array[], int size)
{
    int sum = 0;
    int flag = 0;
    for (int i = 0; i < size; i++)
    {
        if ((flag == 1) || (array[i] == abs_max(array, size)))
        {
            sum = sum + array[i];
            flag = 1;
        }
    }
}
```

```

    }
    return sum;
}

int main()
{
    int array[100];
    int i = 0;
    int menu;
    int size = 0;
    char flag;
    scanf("%d", &menu);
    while (flag != '\n')
    {
        scanf("%d%c", &array[i], &flag);
        i++;
    }
    size = i;

    switch(menu)
    {
        case 0: printf("%d", abs_max(array, size));
        break;
        case 1: printf("%d", abs_min(array, size));
        break;
        case 2: printf("%d", diff(array, size));
        break;
        case 3: printf("%d", sum(array, size));
        break;
        default: printf("Данные некорректны");
    }
    return 0;
}

```

