

**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: Условия, циклы, оператор switch**

Студентка гр. 0382

\_\_\_\_\_

Ситченко К.С.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р

Санкт-Петербург

2020

### **Цель работы.**

Исследовать инициализацию переменных и массивов, а также дальнейшую передачу их в функции. Изучить работу с условным оператором if, оператором switch и циклом for.

### **Задание.**

#### **Вариант 5.**

Напишите программу, выделив каждую подзадачу в отдельную функцию.

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".

### **Выполнение работы.**

Первое число, которое подается программе на вход, записывается в переменную action, значение которой в дальнейшем будет проверяться в операторе switch. Оставшиеся числа записываются в массив arr, а его размер, посчитанный в цикле, сохраняется в переменную size. В зависимости от значения переменной action, выполняются следующие функции:

“0”: abs\_max(int arr[], int n) – на вход подается массив и его размер, в цикле for находится максимальное по модулю число в массиве, затем возвращается его значение.

“1”: `abs_min(int arr[], int n)` – на вход подается массив и его размер, в цикле `for` находится минимальное по модулю число в массиве, затем возвращается его значение.

“2”: `diff(int arr[], int n)` – на вход подается массив и его размер, функция возвращает разницу между максимальным по модулю и минимальным по модулю элементом.

“3”: `sum(int arr[], int n)` – на вход подается массив и его размер, функция возвращает сумму элементов массива, расположенных после максимального по модулю элемента (включая этот элемент).

Если переменная `action` не соответствует ни одному из вышеперечисленных значений, программа выведет сообщение об ошибке.

Разработанный программный код см. в приложении А.

## Тестирование.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1	0 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 -26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9	30	Программа вывела на экран максимальный по модулю элемент
2	0	Данные некорректны	Программа вывела сообщение об ошибке, так как не был введен массив
3	6 7 4 -6 -56 0	Данные некорректны	Программа вывела сообщение об ошибке, так как значение первого входного числа должно быть от 0 до 3
4	1 4 5 -7 -89 0	0	Программа вывела на экран минимальный по модулю элемент
5	2 24 5 -9 -5 -34 56 76 -76	71	Программа вывела на экран разницу между макс. по модулю и мин. по модулю элементом
6	3 4 5 3 87 -54 -20 45 -3 58	113	Программа вывела на экран сумму элементов массива, начиная с максимального

## **Выводы.**

Были изучены основные управляющие конструкции языка Си: цикл *for*, оператор *switch* и условный оператор *if*. Исследована инициализация переменных и массивов, а также дальнейшая передача их в функции.

Разработана программа, выполняющая считывание с клавиатуры исходных данных в виде команды пользователя и массива. Программа обрабатывала команду пользователя с помощью оператора *switch* и условных операторов *if-else*. Для обработки массива в функциях и получения итогового результата использовался условный оператор *if* и функция *abs()* для нахождения модуля числа.

## Приложение А

### Исходный код программы

#### 1. Название файла: main.c

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

int abs_max(int arr[], int n) {
    int abs_max=-1; int i;
    for (i=0; i < n; i++)
        if (abs(arr[i]) > abs(abs_max))
            abs_max = arr[i];
    return abs_max;
}

int abs_min(int arr[], int n) {
    int abs_min=abs(arr[0]); int i;
    for (i=1; i < n; i++)
        if (abs(arr[i]) < abs(abs_min))
            abs_min = arr[i];
    return abs_min;
}

int diff(int arr[], int n) {
    return abs_max(arr, n) - abs_min(arr, n);
}

int sum(int arr[], int n){
    int sum=0; int i, index=0;
    for (i=1; i<n; i++)
        if (abs(arr[i]) > abs(arr[index]))
            index = i;
    for (i=index; i < n; i++)
        sum += arr[i];
    return sum;
}

int main(void){
    int action, size=0;
    char s;
    int arr[100];
    scanf("%d%c", &action, &s);
    while (size <= 100 && s == ' '){
        scanf("%d%c", &arr[size], &s);
        size++;
    }

    if (size == 0)
        printf("Данные некорректны\n");
    else{
        switch (action){
            case 0:
                printf("%d\n", abs_max(arr, size));
                break;
            case 1:
                printf("%d\n", abs_min(arr, size));
                break;
        }
    }
}
```

```
        break;
    case 2:
        printf("%d\n", diff(arr, size));
        break;
    case 3:
        printf("%d\n", sum(arr, size));
        break;
    default:
        printf("Данные некорректны\n");
        break;
    }
}
return 0;
}
```