

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

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Программирование»**  
**Тема: Условия, циклы, оператор switch**

Студентка гр. 2384

Валеева А. А.

Преподаватель

Гаврилов А. В.

Санкт-Петербург

2022

## Цель работы.

Изучить управляющие конструкции языка Си. Получить практические навыки работы с условиями, циклами, оператором `switch`.

## Задание.

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.

Подключаются заголовочные файлы `<stdio.h>` и `<stdlib.h>`, второй необходим для возможности использования функции нахождения модуля (*abs()*).

Используется директива `#define` для определения константы `LIMIT_SIZE = 100` — числа, обозначающего максимальную длину введенного массива. Перед функцией *main* объявляются и описываются четыре функции:

```
int abs_max(int array[], int size);
int abs_min(int array[], int size);
int diff(int array[], int size);
int sum (int array[], int size);
```

Разберем функцию *main*.

Вводим переменную *current\_size*, которая примет значение длины введенного пользователем массива.

Пользователь вводит номер операции от 0 до 3, значение заносится в переменную *value* типа *int*.

Далее в цикле *do... while*:

1. Вводится массив, пока не будет введен символ переноса строки `\n` (прописано условие, равна ли введенная переменная *symbol* типа *char* символу «`\n`») или количество элементов массива не дойдет до 100 (*LIMIT\_SIZE*).

2. Увеличиваем переменную *current\_size* типа *int*, принимающую значения длины массива, на 1.

Используем условный оператор *switch* для вызова функции в функции *printf* в зависимости от значения введенной переменной *value* типа *int*.

При значении *value* = 0:

Функция *abs\_max*, на вход которой подается целочисленный массив *array* и переменная *size* типа *int*, возвращает значение максимального по модулю элемента. Его поиск осуществляем с помощью цикла *for*, в котором по индексам перебираются элементы массива. Перебор начинается с 1, так как значение нулевого элемента массива было принято как максимум, с модулем которого мы будем сравнивать модули следующих элементов. Заканчивается перебор тогда, когда индекс превысит значение переменной *current\_size* типа *int*, содержащей длину массива, введенного пользователем. Телом цикла является условие *if*, в котором сравниваются модули максимального и текущего элемента:

```
if (abs(array[index]) > abs(max_arr)){
    max_arr = array[index];
}
```

При значении *value* = 1:

Аналогично функции *abs\_max*, функция *abs\_min* возвращает значение минимального по модулю элемента массива. Отличие функций заключается в условии оператора *if*:

```
if (abs(array[index]) < abs(min_arr)){
    min_arr = array[index];
}
```

При значении *value* = 2:

На вход функции *diff* подается массив *array* и переменная *size* типа *int*. Переменной *max* типа *int* присваиваю значение функции *abs\_max* от массива *array* и переменной *size* типа *int*, а переменной *min* типа *int* - значение функции *abs\_min* от тех же переменных.

Функция возвращает разность между переменными *max* и *min*.

При значении *value* = 3:

Функция *sum*, на вход которой подается массив *array* и переменная *size* типа *int*, возвращает сумму чисел, находящихся после первого максимального по модулю элемента массива, включая его значение. Введем переменные типа *int* - *max\_index*, который будет содержать в себе индекс первого максимального по модулю элемента массива, и *sum*, переменная, в которой будет высчитана искомая сумма. С помощью цикла *for* перебираем элементы массива, сравнивая модуль максимального и текущего элемента. При подходящем условии переменная *max\_index* будет принимать значение переменной *index* типа *int*, содержащей индекс текущего элемента.

```
for (int index = 1; index < size; index++){
    if (abs(array[index]) > abs(max_arr)){
        max_arr = array[index];
        max_index = index;
    }
}
```

Второй цикл *for* перебирает массив, начиная с переменной *max\_index*. Прибавляем к переменной *sum* текущий элемент.

```
for (int j = max_index; j < size; j++){
    sum = sum + array[j];
}
```

Таким образом, оператор *switch* выводит значение одной из четырех функций с 0 до 3 либо «Данные некорректны», если пользователем было введено что-либо другое, кроме 0, 1, 2, 3.

Разработанный программный код см. в приложении А.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

<i>value</i>	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	Верный ответ
2.	1 4 88 99 77 6 4 -50 -100	4	Верный ответ
3.	2 89 -67 89 3 8 6 -87 67 5	86	Верный ответ
4.	3 98 66 44 5 34 8 94 3	352	Верный ответ
5.	5 88 6 5 4 3 98 76 4	Данные некорректны	Нет команды под номером 5

## Выводы.

Были изучены управляющие конструкции языка Си. Получены практические навыки работы с циклами, условиями и оператором *switch*.

Была разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя и выводящая ответ на один из его запросов. Для обработки команд пользователя использовался условный оператор *switch*, выполняющий различные сценарии при помощи вызова функций, для выполнения каждой из задач — отдельная функция, условный оператор *if*, циклы *for*, *while*.

# ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Valeeva\_Alina\_lb1.c

```
#include <stdio.h>
#include <stdlib.h>
#define LIMIT_SIZE (100)

int abs_max(int array[], int size){
    int max_arr = array[0];
    for (int index = 1; index < size; index++){
        if (abs(array[index]) > abs(max_arr)){
            max_arr = array[index];
        }
    }
    return max_arr;
}

int abs_min(int array[], int size){
    int min_arr = array[0];
    for (int index = 1; index < size; index++){
        if (abs(array[index]) < abs(min_arr)){
            min_arr = array[index];
        }
    }
    return min_arr;
}

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

int sum(int array[], int size){
    int max_arr = array[0];
    int max_index = 0;
    int sum = 0;
    for (int index = 1; index < size; index++){
        if (abs(array[index]) > abs(max_arr)){
            max_arr = array[index];
            max_index = index;
        }
    }

    for (int j = max_index; j < size; j++){
        sum = sum + array[j];
    }

    return sum;
}

int main(){
    int value;
    int arr[LIMIT_SIZE];
    int current_size = 0;
```

```

char symbol;
scanf("%d%c", &value, &symbol);
do{
    scanf("%d%c", &arr[current_size ++], &symbol);
    }while(symbol != '\n' && current_size < LIMIT_SIZE);

switch (value){
    case 0:
    {
        printf("%d\n", abs_max(arr, current_size));
        break;
    }
    case 1:
    {
        printf("%d\n", abs_min(arr, current_size ));
        break;
    }
    case 2:
    {
        printf("%d\n", diff(arr, current_size));
        break;
    }
    case 3:
    {
        printf("%d\n", sum(arr, current_size));
        break;
    }
    default:
    {
        printf("Данные некорректны");
        break;
    }
}
}

```