

**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**

Студент гр. 0382

\_\_\_\_\_

Куликов М.Д.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2020

## **Цель работы.**

Изучение базовых конструкций языка Си.

## **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".

## **Основные теоретические положения.**

При выполнении данной программы были использованы:

### **Стандартные функции:**

printf () – выводит заданное значение или элемент на экран

scanf () – считывает входные данные

### **Циклы:**

while () – блок кода после условия цикла выполняется до тех пор, пока условие в круглых скобках цикла является истинным.

for () – в качестве первого аргумента цикла принимается переменная цикла с изначальным значением, во втором аргументе пишется условие, в течение которого выполняется данный цикл, в качестве третьего

аргумента обычно указывают выражение, увеличивающее переменную цикла.

### **Операторы:**

`if()` – условный оператор, который начинает выполнение блока кода после него, если выполняется определенное условие, переданное в качестве аргумента оператора. Иначе, выполнится блок кода после оператора `else`, если таковой имеется

`switch(x)` – принимает в качестве аргумента переменную. После этого выполняет блок кода, следующий за «`case x :`» с переданным значением переменной. Если значение переменной не описано не в одном «`case`», то тогда выполняется блок кода, следующий после «`default:`».

### **Библиотеки:**

**<stdlib.h> - была включена ради объявления функции `abs()`**

**<stdio.h>**

### **Выполнение работы.**

В самом начале вводится константа `max_amount`, с помощью которой можно регулировать максимально допустимое число элементов массива. Также добавляются библиотеки.

Далее описываются 4 функции, которые понадобятся в дальнейшем для использования в операторе `switch`.

1) `abs_max()` - первым аргументом данной функции является массив, вторым — количество элементов в нем. С помощью цикла `for` мы перебираем этот массив и ищем элемент в максимальным значением по модулю и его индекс. Функция возвращает индекс максимального по модулю числа в массиве.

2) `abs_min()` - эта функция является копией прошлой функции за исключением того, что она ищет минимальный по модулю элемент и его индекс в массиве.

3) `diff()` - первым аргументом данной функции является массив, вторым — количество элементов в нем. С помощью функций `abs_max` и `abs_min` мы ищем

разность между максимальным и минимальным по модулю элементом. Функция возвращает значение этой разности.

4) `sum()` - первым аргументом данной функции является массив, вторым — количество элементов в нем. С помощью функции `abs_max` и цикла `for` мы складываем значение всех элементов массива, которые идут после максимального, включая максимальный элемент. Функция возвращает сумму этих элементов.

Далее идет описание функции `main`:

Мы объявляем некоторые переменные:

`int choice`- выбор опции пользователем, которая будет использована в операторе `switch`

`int arr[]` - объявляем массив, в котором будут храниться введенные пользователем числа

`int arr_size` — количество заполненных ячеек массива

`char sym` — изначально этой переменной присваивается «пробел».

Делается это для того, чтобы при вводе массива в массив вводилось только число.

Пользователь вводит цифру, опцию которой он хочет выбрать.

После этого мы начинаем заполнение массива:

Если количество элементов массива не превышает максимальный и переменная `sym == « »`, тогда считывается сначала число массив, а потом символ, который должен являться пробелом.

Затем мы вызываем оператор `switch`:

Если пользователь ввел 0 — на экран выводится элемент массива с индексом `abs_max`.

Если пользователь ввел 1 -на экран выводится элемент массива с индексом `abs_min`.

Если пользователь ввел 2 — на экран выводится разница между максимальным и минимальным по модулю элементом массива.

Если пользователь ввел 3 — на экран выводится сумма элементов массива после максимального по модулю, включая максимальный

Если переменная choice имеет другое значение — на экран выводится сообщение «Данные некорректны»

Выполнение функции main заканчивается, возвращается 0.

### **Тестирование.**

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 3 5 6 848 -938 -5 87 -4	-938	abs_max
2.	1 4564 0 93857 -875 -948 784	0	abs_min
3.	2 1 785 87 -45 876	875	diff
4	3 2 3 4 5 1 1 1 1	9	sum
5	5 6 5 7 3 5	Данные некорректны	error

### **Выводы.**

В ходе работы были изучены базовые конструкции языка Си.

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. Входные данные были отсортированы и введены в массив с помощью цикла while и функции scanf. Далее вызывалась функция, предопределенная выбором пользователя и на экран выводилась информация, которую он запросил. Это было реализовано с помощью пользовательских функций и оператора switch. При некорректном выборе опции на экран выводилась строка «Данные некорректны».

# ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab1.c

```
#include <stdio.h>
#include <stdlib.h>
#define max_amount 100

int abs_max(int numbers[], int quantity){
    int maxnum = abs(numbers[0]);
    int maxindex = 0;
    for (int i = 1; i < quantity; i++){
        if( abs(numbers[i]) > abs(maxnum)){
            maxnum = abs(numbers[i]);
            maxindex = i;
        }
    }

    return maxindex;
}

int abs_min(int numbers[], int quantity){
    int minindex = 0;
    int minnum = abs(numbers[0]);
    for (int i = 1; i < quantity; i++){
        if( abs(numbers[i]) < minnum){
            minnum = abs(numbers[i]);
            minindex = i;
        }
    }

    return minindex;
}

int diff (int numbers[], int quantity){
    int diff = ( numbers[abs_max(numbers, quantity)] -
numbers[abs_min(numbers, quantity)]);
    return diff;
}

int sum (int numbers[], int quantity){
    int sum = 0;
    for(int i = abs_max(numbers, quantity); i < quantity; i++){
        sum += numbers[i];
    }
    return sum;
}

int main(){

    int arr[max_amount], arr_size = 0, choice;
    char sym = ' ';
    scanf("%d", &choice);
    while (arr_size < max_amount && sym == ' '){
        scanf("%d%c", &arr[arr_size++], &sym);
    }
}
```

```

    }
    switch(choice) {
    case 0:
        printf("%d\n", arr[abs_max(arr,arr_size)]);
        break;
    case 1:
        printf("%d\n", arr[abs_min(arr,arr_size)]);
        break;
    case 2:
        printf("%d\n",diff (arr,arr_size));
        break;
    case 3:
        printf("%d\n",sum (arr,arr_size));
        break;

    default:
        printf("Данные некорректны \n");

    }
    return 0;
}

```