

**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. 0383

Шквиря Е.В.

Преподаватели

Чайка К.В.  
Жангиров Т.Р.

Санкт-Петербург

2020

### **Цель работы.**

Изучить устройство условных операторов, циклов и оператора *switch* в языке Си.

### **Задание.**

Вариант 2.

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 число в массиве. (*max*)

1: Минимальное число в массиве. (*min*)

2: Разницу между максимальным и минимальным элементом. (*diff*)

3: Сумму элементов массива, расположенных до первого минимального элемента. (*sum*)

иначе необходимо вывести строку «Данные некорректны».

### **Выполнение работы.**

Порядок выполнения поставленной задачи программой:

1. Считывание массива чисел. Запись чисел в массив производится до первой встречи в вводе символа переноса строки, что проверяется в условии *while*. Инициализация массива числами производится после ввода каждого из них.

2. Оператор *switch* используется для определения команды, которая будет выполняться над массивом. Если команда является правильной и предусмотренной программой, то будет произведён вывод результата

соответствующей функции обработки массива. Иначе, будет выведен результат «Данные некорректны».

3. Каждая подзадача, требуемая условием задачи, реализована в качестве отдельной функции (*min*, *max*, *diff*, *sum*), возвращающей результат обработки массива. Изначально указаны определения функций, а в дальнейшем их объявление.

Используемые переменные:

MAX\_SIZE — максимальный размер массива.

SIZE\_ERROR — ошибка, связанная с нулевым размером массива.

END\_CHAR — символ, являющийся признаком конца строки.

command — команда, запрашиваемая пользователем.

array — массив чисел, вводимый пользователем.

size — размер массива array. Вычисляется по ходу считывания чисел.

inputVar — временная переменная, в которую мы постепенно записываем результат обработки символов для одного числа.

tempChar - временная переменная, в которую мы считываем символ из потока ввода.

maxElement/minElement — максимальный/минимальный элемент массива.

sumAnswer — сумма элементов до первого минимального числа в массиве.

Используемые функции:

max — поиск максимального элемента в массиве.

min — поиск минимального элемента в массиве.

diff — вычисление разности между максимальным и минимальным элементом массива.

sum — вычисление суммы элементов до первого минимального числа в массиве.

`main` — функция, являющаяся началом программы.

Разработанный программный код см. в приложении А.

### Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 1 -2 3 -4 5 -6	5	Верный ответ
2.	1 -60 59 2 -4 -89	-89	Верный ответ
3.	2 -2 -3 -1 3 1 2	6	Верный ответ
4.	3 50 100 -100 -50	150	Верный ответ
5.	3 10	0	Верный ответ
6.	1	Данные некорректны	Верный ответ
7.	-10 1 2 3	Данные некорректны	Верный ответ

### Выводы.

Было изучено устройство условных операторов, циклов и оператора *switch* в языке Си.

Разработана программа, считывающая с клавиатуры массив чисел и позволяющая найти в нём минимальный и максимальный элемент, разность между ними и сумму элементов до первого минимального элемента. Для обработки команд пользователя использовались оператор *switch*, циклы и условные операторы *if-else*. Для считывания данных из потока ввода использовались функции *getchar()* и *scanf()*, для вывода результата использовалась функция *printf()*.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab1.c

```
#include <stdio.h>

const int MAX_SIZE = 100;
const int SIZE_ERROR = -1;
const char END_CHAR = '\n';

int max(int array[], int size);

int min(int array[], int size);

int diff(int array[], int size);

long sum(int array[], int size);

int main()
{
    int command = 0;
    scanf("%d", &command);
    int array[MAX_SIZE];
    int size = 0;
    int inputVar = 0;
    char tempChar = '_';

    while((tempChar = getchar()) != END_CHAR)
    {
        scanf("%d", &inputVar);
        array[size++] = inputVar;
    }
}
```

```

//проверка для пустого массива
if(size == 0)
    command = SIZE_ERROR;

switch (command)
{
    case 0://Максимальное число
        printf("%d\n", max(array, size));
        break;

    case 1://Минимальное число
        printf("%d\n", min(array, size));
        break;

    case 2://Разность между максимальным и минимальным значением
        printf("%d\n", diff(array, size));
        break;

    case 3://Сумма чисел до первого минимального элемента
        printf("%ld\n", sum(array, size));
        break;

    default://Некорректные данные
        printf("Данные некорректны\n");
}

return 0;
}

```

```

int max(int array[], int size)
{
    int maxElement = array[0];
    for (int i = 1; i < size; ++i)
    {
        if (array[i] > maxElement)

```

```

        maxElement = array[i];
    }
    return maxElement;
}

int min(int array[], int size)
{
    int minElement = array[0];
    for (int i = 1; i < size; ++i)
    {
        if (array[i] < minElement)
            minElement = array[i];
    }
    return minElement;
}

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

long sum(int array[], int size)
{
    int minElement = min(array, size);
    long sumAnswer = 0;
    for (int i = 0; i < size; ++i)
    {
        if (array[i] != minElement)
            sumAnswer += array[i];
        else
            return sumAnswer;
    }
}

```