

**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. 9304		Мохаммед А. А.
Преподаватель		Чайка К. В.

Санкт-Петербург
2019

Цель работы.

Изучить синтаксис управляющих конструкций языка Си (циклы, условия, оператор switch), массивы и функции, применить знания на практике.

Задание.

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: сумма элементов массива, расположенных до первого минимального элемента. (sum)

иначе необходимо вывести строку «Данные некорректны».

Основные теоретические положения.

Массив.

Массив - структура данных, в виде набора элементов одного типа, расположенные в памяти строго друг за другом.

Объявление массива: `тип`

`имя[размер];`

Функция.

Функция – это самостоятельная единица программы, которая создана для реализации конкретной подзадачи. Функция является подпрограммой, которая

может содержаться в основной программе, а может быть создана отдельно (в библиотеке). Каждая функция выполняет в программе определенные действия.

Условный оператор.

Условный оператор (if-else) – оператор проверки условия. Имеет полную (с else) и неполную (без else) форму. Если условие истинно, выполняется тело, идущее после if, иначе выполняется тело, идущее после else. Если else отсутствует и условие ложно, тело условного оператора пропускается, и программа продолжает выполнение операций.

Синтаксис полного условного оператора:

```
if (условие){  
    телоif;  
} else {  
    тело else;  
}
```

Синтаксис неполного условного оператора:

```
if (условие) {  
    тело;  
}
```

Оператор switch.

Switch – оператор множественного выбора. В зависимости от значения поступающей переменной оператор передает управление соответствующей ветке. Если не один из вариантов не подходит, выполнится тело, находящееся в ветке default.

Выполнение работы.

Подключил библиотеки:

```
#include<stdio.h>#  
include<limits.h>#  
include<locale.h>
```

Представим что 10000 символов хватит для чтения потока и зададим максимальный размер массива 100 (из условия)

```
#defineINPUT_SIZE10000  
#defineNOTHING-1
```

Далее идут функции (см. Реализованные функции)

После чего создаем буффер для чтения символов из входного потока

```
charbuffer[INPUT_SIZE]
```

Считываем в буффер символы входного потока

```
fgets(buffer,sizeof(buffer),stdin)
```

Создаем указатель для перемещения по буферу

```
char* data = buffer
```

Задаем переменную для хранения размера массива

```
int n = 0
```

Задаем переменную для хранения сдвига по входному потоку

СИМВОЛОВ

```
intoffset
```

Первое число считывается для выбора пункта меню, если чтение удалось, идем дальше

```
if(sscanf_s(data,"%d\n", &number, &offset) == 1)
```

Записываем число как ответ для выбора пункта меню

```
answer = number
```

Сдвигаемся в потоке символов на прочтенную длину

```
data += offset
```

Считываем новые числа до тех пор, пока не встретим конец или неподходящие символы

```
while(sscanf_s(data, " %d%n", &number, &offset) == 1)
```

Записываем считанное число в массив

```
arr[n] = number
```

Увеличиваем логический размер кластера

```
n++
```

Сдвигаемся в потоке символов на прочтенную длину

```
data += offset
```

Выполняем действие по выбору

```
switch(answer)
```

```
case 0: Вывести максимально число в массиве
```

```
printf("%d\n", Max(arr,  
n));break
```

```
1: Вывести минимальное число в массиве
```

```
printf("%d\n", Min(arr,  
n));break;
```

```
2: Вывести          разницу между максимальным и  
минимальным элементом
```

```
printf("%d\n", Max(arr, n) -  
Min(arr,n));break;
```

```
3: Вывести сумму элементов массива, расположенных до  
первого минимального элемента
```

```
printf("%d\n",  
SumArrayElementsInRangeFromZeroToIndex(FindIndexFirstMin  
imum(arr,n),arr));  
break;
```

В случае ошибки вывести

```
default:  
    printf("Данные некорректны");
```

Так же установил русскую локализацию для корректной работы кириллицы в консоли

```
setlocale(LC_ALL, "Rus");
```

- Реализованные функции:
 - `int Max(int arr[], int n)` для нахождения максимального элемента в массиве. использовалась
 - `int Min(int arr[], int n)` для нахождения минимального элемента в массиве. использовалась
 - `int FindIndexFirstMinimum(int arr[], int n)` использовалась для нахождения индекса первого минимального элемента массива
 - `int SumArrayElementsInRangeFromZeroToIndex(int end, int arr[])` использовалась для вычисления суммы чисел находящихся в диапазоне от начала до определенного индекса

Разработанный программный код см. в приложении.

Результаты тестирования.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования.

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 23 131 232 31 94	232	Тест пройден
2.	3 142 232 19 -7132	393	Тест пройден
3.	1 31 -237 11 232 149	-237	Тест пройден
4.	2 942 42 12 32 54 10	932	Тест пройден
5.	1 23 0 -231 23 -11 35	-231	Тест пройден
6.	0 123 233 201 334 335	335	Тест пройден
7.	6 12 40 959 42 246 22	Данные некорректны	Тест пройден
8.	0 31 93 44 12 47 -94	93	Тест пройден
9.	2 331 -39 40 112 32 9	370	Тест пройден
10.	4 223 11 23 15 3	Данные некорректны	Тест пройден
11.	2 23 132 532 1 56	531	Тест пройден

12.	0 221 131 67 48	221	Тест пройден
-----	-----------------	-----	--------------

Выводы.

В ходе лабораторной работы были изучены: структура программы языка программирования C, массивы элементов, функции и основные управляющие конструкции языка if-else, switch, while, for. Была разработана программа, которая считывает с клавиатуры исходные данные, формирует массив из исходных данных и выполняет определённые операции над этими данными. Для обработки команд пользователя использовались условные операторы if-else, операторы циклов for.

ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Мохаммед А.А_pr_lb1

```
#include <stdio.h>
#include <limits.h>
#include <locale.h>
```

```
#define INPUT_SIZE 10000
#define NOTHING -1
```

```
int arr[100];
```

```
int Max(int arr[], int n){
    int max = INT_MIN;
    for (int index = 0; index < n; index++){
        if (arr[index] > max)
            {max = arr[index];}
    }
    return max;}
```



```

int Min(int arr[], int n){
    int min = INT_MAX;
    for (int index = 0; index < n; index++){
        if (arr[index] < min){
            min = arr[index];
        }
    }
    return min;
}

```

```

int FindIndexFirstMin(int arr[], int n){
    int min = Min(arr, n);
    for (int index = 0; index < n; index++){
        if (arr[index] == min){
            return index;
        }
    }
    return NOTHING;
}

```

```

int SumIndex(int end, int arr[]){
    int sum = 0;
    for (int index = 0; index < end; index++){
        sum += arr[index];
    }
    return sum;
}

```

```

int main(){
    char buffer[INPUT_SIZE];

    fgets(buffer, sizeof(buffer), stdin);
    char* data = buffer;
    int n = 0;
    int t = NOTHING;
    int number;
    int x;

    if (sscanf(data, "%d%n", &number, &x) == 1){
        t = number;
        data += x;
        while (sscanf(data, "%d%n", &number, &x) == 1){
            arr[n] = number;
            n++;
        }
    }
}

```

```
data += x;}}
```

```
switch (t) {  
case 0: {  
printf("%d\n", Max(arr, n));  
break;}  
case 1: {  
printf("%d\n", Min(arr, n));  
break;}  
case 2: {  
printf("%d\n", Max(arr, n) - Min(arr, n));  
break;}  
case 3: {  
printf("%d\n", SumIndex(FindIndexFirstMin(arr, n), arr));  
break;}  
default:  
printf("Данные некорректны");}  
  
return 0;}
```