

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)  
Кафедра МО ЭВМ**

**ОТЧЕТ  
по лабораторной работе №2  
по дисциплине «Программирование»  
Тема: Условия, циклы,  
оператор switch**

Студент гр. 9304		Мохаммед А. А.
Преподаватель		Чайка К. В.

Санкт-  
Петербург 2019

### Цель работы.

Научиться создавать make-файл, изучить работу препроцессора, компилятора, линковщика.

### Задание.

В текущей директории создайте проект с make-файлом. Главная цель должна приводить к сборке проекта. Файл, который реализует главную функцию, должен называться `menu.c`; исполняемый файл - `menu`. Определение каждой функции должно быть расположено в отдельном файле, название файлов указано в скобках описания каждой функции.

Реализуйте функцию-меню, 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:

- : максимальное число в массиве. (`max.c`)
- : минимальное число в массиве. (`min.c`)
- : разницу между максимальным и минимальным элементом. (`diff.c`)
- : сумму элементов массива, расположенных до минимального элемента. (`sum.c`)

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: `ти`

`пимя[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.

### **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

## **Выводы.**

Была изучена работа компилятора, препроцессора, линковщика, так же был изучен процесс создания Make-файла.

## **ПРИЛОЖЕНИЕ А**

### **ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: Makefile

```
all: menu.o max.o min.o diff.o sum.o
    gcc menu.o max.o min.o diff.o sum.o -o menu
menu.o: menu.c diff.h sum.h max.h min.h
    gcc -c menu.c sum.o:
sum.c sum.hmin.h
    gcc -c sum.c
diff.o: diff.c diff.h max.h min.h gcc
    -c diff.c
min.o: min.cmin.hgcc -
    cmin.c
max.o: max.cmax.hgcc -
    cmax.c
clean:
    rm -rf *.o menu
```

Название файла: menu.c

```
#include<stdio.h>
#include<limits.h>
#include<locale.h>
#include<max.h>
#include<min.h>
#include<sum.h>\
#include<diff.h>
```

```
#define INPUT_SIZE 10000
```

```
#define NOTHING -1
```

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

```
int main(){
char buffer[INPUT_SIZE];

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

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", diff_between_max_and_min(arr, diff, n));  
break;}  
case 3: {  
printf("%d\n", sum_before_min(arr, n));  
break;}  
default:  
printf("Данные некорректны");}  
  
return 0;}
```

Название файла: max.c

```
#include<stdio.h>  
  
#include"max.h"  
  
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;}
```

Название файла min.c:

```
#include<stdio.h>  
#include"min.h"  
  
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; }
```

Название файла: diff.c

```
#include<stdio.h>  
#include "diff.h"  
#include "max.h"  
#include "min.h"
```

```
int diff_between_max_and_min(int*arr,int diff, int n){  
diff = Max(arr, n) - Min(arr, n);  
return diff;  
  
}
```

Название файла: sum.c:

```
#include<stdio.h>  
#include "sum.h"  
#include "min.h"
```

```
int sum_before_min(int *arr, int n){  
int sum = 0;  
for(int i= 0; arr[i] >Min(arr,n);i++){  
sum=sum + arr[i];}  
return sum;}
```

Название файла: max.h

```
#include<stdio.h>  
  
int Max(int *arr, int n);
```

Название файла: min.h

```
#include<stdio.h>

int Min(int *arr, int n);
```

Название файла: diff.h

```
#include<stdio.h>
int diff_between_max_and_min(int*arr,int diff, int n);
```

Название файла: sum.h

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

```
int sum_befor_min(int *arr, int n);
```