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

ОТЧЕТ

по лабораторной работе №2 по дисциплине «Программирование»

Тема: "Условия, циклы, оператор switch"

Студент гр. 7381	 Габов Е.С
Преподаватель	 Берленко Т.А

Санкт-Петербург 2017

Целью данной работы является:

- 1. Научиться пользоваться оператором выбора Switch и условным оператором While
- 2. Отточить навыки создания Makefile.
- 3. Создания программы, которая на выбор пользователя выполняет одну функций.
- 4. Реализовать 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:
 - 0 : индекс первого отрицательного элемента.
 - 1 : индекс последнего отрицательного элемента.
 - 2 : Найти сумму модулей элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент).
 - 3: Найти сумму модулей элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент).

Иначе выводить строку: «Данные некорректны».

Основные теоретические положения:

В данной лабораторной работе я сначала создал файл menu.c. В этой функции пользователю необходимо вводить номер операции и массив целых чисел с помощью функции scanf (является процедурой ввода общего назначения, считывающей данные из потока stdin. Она может считывать данные всех базовых типов и автоматически конвертировать их в нужный внутренний формат).

Введенный номер операции поступает в функцию выбора switch (Оператор сравнивает значение одной переменной с несколькими константами) .Там он следует одной и 4 инструкций (вызывает нужную функцию):

0)индекс первого отрицательного элемента

- 1)индекс последнего отрицательного элемента
- 2)сумму между отрицательными элементами
- 3) сумму до и после отрицательных элементов

Иначе выводить строку: «Данные некорректны».

Функция возвращает значение и в операторе switch выводит результат на экран.

Для того чтобы заголовочные файлы не подключались только один раз использовал код

```
#ifndef <name>
#define <name>
<type> <name>
#endif
```

В функциях sum_between_negative.c and sum_before_and_after_negative.c вызывал функции index_first_negative.c and index_last_negative.c для того чтобы избежать ошибки дублирования кода.

Далее создал Makefile в котором скомпилировал программу и назвал исполняемый файл menu с помощью ключа "-o".

```
*Исходный код программы*
****** Makefile *****
all: menu.o index_first_negative.o index_last_negative.o sum_before_and_after_negative.o
sum_between_negative.o
      gcc menu.o index_first_negative.o
index_last_negative.osum_before_and_after_negative.o sum_between_negative.o -o menu
menu.o: menu.c index_first_negative.h index_last_negative.h sum_before_and_after_negative.h
sum_between_negative.h
      gcc -c menu.c
index_first_negative.o: index_first_negative.c index_first_negative.h gcc -c
index_first_negative.c
index_last_negative.o: index_last_negative.c index_last_negative.h
      gcc -c index_last_negative.c
sum_before_and_after_negative.o: sum_before_and_after_negative.c index_first_negative.h
index_last_negative.h sum_before_and_after_negative.h
      gcc -c sum_before_and_after_negative.c
sum_between_negative.o: sum_between_negative.c index_first_negative.h index_last_negative.h
sum between negative.h
      gcc -c sum_between_negative.c
           menu.c ******
*****
#include <stdio.h>
#include "index_first_negative.h"
#include "index_last_negative.h"
#include "sum_before_and_after_negative.h"
#include "sum between negative.h"
int main()
 int i=0;
 int operation_number, res;
 char c;
 int mas[100];
 scanf( "%d%c", &operation_number , &c );
 scanf( "%d%c", &mas[0], &c );
 while (c!='\n')
```

```
{
   i++;
   scanf( "%d%c", &mas[i], &c);
i++;
res = 0;
 switch ( operation_number )
   case 0: res = index_first_negative( mas , i );
        printf("%d", res );
        break;
   case 1: res = index_last_negative ( mas , i );
        printf("%d", res );
        break;
   case 2: res = sum_between_negative ( mas , i );
        printf("%d", res );
        break;
   case 3: res = sum_before_and_after_negative ( mas , i );
        printf("%d", res );
        break;
   default: printf("????????????");
}
```

```
******index_first_negative.c*****
#include "index_first_negative.h"
int index_first_negative( int mas[] , int size )
{
   int i=0;
   while ( i < size )
   {
      if ( mas[i] < 0 )
        {
        return i;
      i=i+101;
      }
   i++;
   }
}</pre>
```

```
***** index last negative ******
#include "index_last_negative.h"
int index_last_negative ( int mas[] , int size )
{
 int i=1:
 while (i < size)
   if (mas[size-i] < 0)
        return size-i;
        i=i+100;
   i++;
}
******sum_before_and_after_nagative.c*****
#include "sum_before_and_after_negative.h"
#include "index_first_negative.h"
#include "index_last_negative.h"
#include <stdlib.h>
int sum_before_and_after_negative(int mas[], int size)
 int first, last;
 int i = 0;
 int sum = 0;
 first = index_first_negative ( mas , size );
 while (i < first)
   sum+=mas[i];
   i++;
 i = index_last_negative ( mas , size );
 while (i < size)
   sum+=abs(mas[i]);
   i++;
 return sum;
}
******sum between negative.c*****
#include "sum_between_negative.h"
#include "index_first_negative.h"
#include "index_last_negative.h"
int sum_between_negative( int mas[] , int size )
```

```
int first, last;
int sum=0;
 int i=0;
 first=index_first_negative( mas , size );
 last=index_last_negative( mas , size );
 i = first;
 while (i < last)
   sum+=abs(mas[i]);
   i++;
return sum;
*****index first negative.h****
#ifndef __INDEX_FIRST_NEGATIVE_H__
#define __INDEX_FIRST_NEGATIVE_H__
int index_first_negative(int [], int);
#endif
*******index last negative.h*****
#ifndef __INDEX_LAST_NEGATIVE_H__
#define __INDEX_LAST_NEGATIVE_H__
int index_last_negative(int [], int);
#endif
****sum_before_and_after_negative.h*****
#ifndef __SUM_BEFORE_AND_AFTER_NEGATIVE_H_
      #define __SUM_BEFORE_AND_AFTER_NEGATIVE_H__
      int sum_before_and_after_negative ( int [], int );
      #endif
*****sum between negative******
#ifndef __SUM_BETWEEN_NEGATIVE_H__
#define SUM BETWEEN NEGATIVE H
int sum_between_negative(int [], int);
#endif
```