**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 c помощью ключа “-о”.

\*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++;

}

}

\*\*\*\*\*\*\* 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