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 _____ Павлов А.П.

Преподаватель _____ Берленко Т.А.

Санкт-Петербург
2017

Цель работы.

Научиться работать с массивами, освоить работу с оператором выбора switch, с циклами for (;;), while (), do while(),а также с операторами case, break и default.

В текущей директории создать 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 около описания каждой функции.

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 нулевого элемента. (index_first_zero.c)

1: индекс последнего нулевого элемента. (index_last_zero.c)

2: Найти сумму модулей элементов массива, расположенных от первого нулевого элемента и до последнего. (sum_between.c)

3: Найти сумму модулей элементов массива, расположенных до первого нулевого элемента и после последнего. (sum_before_and_after.c)

иначе необходимо вывести строку "Данные некорректны".

Основные теоретические положения.

Заголовочные файлы, необходимые для создания проекта:

1. <stdio.h> - содержит прототипы функций "int printf(const char* format [, argument]...);" и "int scanf(const char* format [, argument]...);", которые используются для ввода из потока ввода и вывода в поток вывода.

Синтаксис:

#include < stdio.h > int printf (const char *format, ...);

Аргументы:

format – указатель на строку с описанием формата.

Возвращаемое значение:

При успешном завершении вывода возвращается количество выведенных символов. При ошибке возвращается отрицательное число.

Описание:

Функция printf выводит в стандартный поток вывода строку отформатированную в соответствии с правилами, указанными в строке, на которую указывает аргумент format.

Правила задаются набором трех типов директив: 1. Обычные символы (кроме '%' и '\'), которые выводятся без изменения; 2. Спецификаторы формата; 3. Специальные сиволы.

Синтаксис:

#include < stdio.h > int scanf(const char *format, ...);

Аргументы:

format – указатель на строку с описанием формата.

Возвращаемое значение:

Возвращает число, равное количеству полей, значения которых были действительно присвоены переменным. Если до присвоения значения первого поля произошла ошибка, возвращается EOF.

Описание:

Функция scanf() является процедурой ввода общего назначения, считывающей данные из потока stdin.

2. <stdlib.h> - содержит прототип функции "int abs(int n);", возвращающей абсолютное значение числа.

Синтаксис:
#include < string.h > int abs(int n);

Аргументы:

n – целое значение.

Возвращаемое значение:

Возвращает абсолютное значение числа.

Вывод:

В ходе выполнения лабораторной работы были освоены основы работы с массивами, оператор выбора switch, циклы for (;;), while (), do while(),а также операторы case, break и default.

Исходный код проекта:

Файл index_first_zero.c

```
int index_first_zero(int arr[], int i)
{
   int first;
   for (first = 0; first <= i; first++)
   {
      if (arr[first] == 0)
         break;
   }
   return first;
}
```

Файл index_first_zero.h

```
#pragma once
int index_first_zero(int arr[], int);
```

Файл index_last_zero.c

```
int index_last_zero(int arr[], int i)
{
   int last;
   for (last = i-1; last >= 0; last--)
   {
      if (arr[last] == 0)
         break;
   }
   return last;
}
```

Файл index_last_zero.h

```
#pragma once
int index_last_zero(int arr[], int);
```

Файл sum_before_and_after.c

```
#include <stdlib.h>
#include "index_first_zero.h"
#include "index_last_zero.h"

int sum_before_and_after(int arr[], int i)
{
   int first, last;
   first = index_first_zero(arr, i) - 1;
   last = index_last_zero(arr, i) + 1;
   int k, sum3, sum1 = 0, sum2 = 0;
   for (k = 0; k <= first; k++)
   {
      sum1 = sum1 + abs(arr[k]);
   }
   for (k = i; k >= last; k--)
   {
      sum2 = sum2 + abs(arr[k]);
   }
   sum3 = sum1 + sum2;
   return sum3;
}
```

Файл sum_before_and_after.h

```
#pragma once
int sum_before_and_after(int arr[], int);
```

Файл sum_between.c

```
#include <stdlib.h>
#include "index_first_zero.h"
#include "index_last_zero.h"

int sum_between(int arr[], int i)
{
   int first, last;
   first = index_first_zero(arr, i)+1;
   last = index_last_zero(arr, i)-1;
   int k, sum1 = 0;
   for (k = first; k <= last; k++)
   {
      sum1 = sum1 + abs(arr[k]);
   }
```

```
    return sum1;
}
```

Файл sum_between.h

```
#pragma once
int sum_between(int arr[], int);
```

Файл menu

```
#include "sum_between.h"
#include "index_first_zero.h"
#include "index_last_zero.h"
#include "sum_before_and_after.h"

int main()
{
    int N, i = 0;
    int arr[100];
    char k;
    scanf("%d", &N);
    do
    {
        scanf("%d%c", &arr[i], &k);
        i++;
    } while (k != '\n');

    switch (N)
    {
    case 0:
        printf("%d", index_first_zero(arr, i));
        break;
    case 1:
        printf("%d", index_last_zero(arr, i));
        break;
    case 2:
        printf("%d", sum_between(arr, i));
        break;
    case 3:
        printf("%d", sum_before_and_after(arr, i));
        break;
    default:
        printf("Данные некорректны");
    }
    return 0;
```

}

Файл Makefile

```
menu: menu.o index_first_zero.o index_last_zero.o sum_between.o
sum_before_and_after.o
	gcc menu.o index_first_zero.o index_last_zero.o sum_between.o
sum_before_and_after.o -o menu
menu.o: menu.c
	gcc -c menu.c
index_first_zero.o: index_first_zero.c index_first_zero.h
	gcc -c index_first_zero.c
index_last_zero.o: index_last_zero.c index_last_zero.h
	gcc -c index_last_zero.c
sum_between_zero.o: sum_between.c sum_between.h
	gcc -c sum_between.c
sum_before_and_after_zero.o: sum_before_and_after.c sum_before_and_after.h
	gcc -c sum_before_and_after.c
clean:
	rm -rf *.o
```