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

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Программирование»

Тема: «Условия, циклы, оператор switch»

Студент гр. 7381 _____ Машина Ю.Д.

Преподаватель _____ Берленко Т. А.

Санкт-Петербург

2017

Цель работы

Познакомиться с оператором выбора switch, с циклами for (;;), while (), do while(),а также 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.

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:

0 : индекс первого отрицательного элемента. (index_first_negative.c)
1 : индекс последнего отрицательного элемента. (index_last_negative.c)
2 : Найти сумму модулей элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент). (sum_between_negative.c)
3 : Найти сумму модулей элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент). (sum_before_and_after_negative.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 результате работы были освоены оператор выбора switch, циклы for, while, do while, операторы case, break и default. Были закреплены знания по созданию проектов с Makefile.

Исходный код проекта

- Файл menu.c

```
#include <stdio.h>
#include <locale.h>
#include "sum_between_zero.h"
#include "index_first_zero.h"
#include "index_last_zero.h"
#include "sum_before_and_after_zero.h"

int main()
{
   setlocale(LC_ALL, "Rus");
   int i = 0;
   int arr[100];
   char t;
   do
   {
      scanf("%d%c", &arr[i], &t);
      i++;
   } while (t != '\n');
   i = i - 1;

   switch (arr[0])
   {
   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_zero(arr, i));
      break;
   case 3:
      printf("%d", sum_before_and_after_zero(arr, i));
      break;
   default:
```

```
        printf("Данные некорректны");
    }
    return 0;
}
```

- Файл index_first_zero.c

```
int index_first_zero(int arr[], int i)
{
    int first;
    for (first = 1; first <= i; first++)
    {
        if (arr[first] == 0)
            break;
    }
    return first - 1;
}
```

- 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; last >= 1; last--)
    {
        if (arr[last] == 0)
            break;
    }
    return last - 1;
}
```

- Файл index_last_zero.h

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

- Файл sum_between_zero.c

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

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

- Файл sum_between_zero.h

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

- Файл sum_before_and_after_zero.c

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

int sum_before_and_after_zero(int arr[], int i)
{
   int first, last;
   first = index_first_zero(arr, i) + 1;
   last = index_last_zero(arr, i) + 1;
   int k, sum2, sum21 = 0, sum22 = 0;
   for (k = 1; k < first; k++)
   {
      sum21 = sum21 + abs(arr[k]);
   }
```

```
    for (k = i; k >= last; k--)
    {
        sum22 = sum22 + abs(arr[k]);
    }
    sum2 = sum21 + sum22;
    return sum2;
}
```

- Файл sum_before_and_after_zero.h

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

- Файл Makefile

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