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

Студентка гр. 7381 Кревчик А.Б.

Преподаватель Берленко Т.А.

Санкт-Петербург

2017

Цель

Ознакомиться с типами данных, функциями стандартной библиотеки ввода/вывода printf и scanf, оператором множественного выбора switch, циклами for(;;), while(), do while().

Задание

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 20. 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 произведение элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент). (multi_between_negative.c)
3 : Найти произведение элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент). (multi_before_and_after_negative.c)

Иначе необходимо вывести строку "Данные некорректны".

Основные теоретические положения

Оператор множественного выбора switch - выполняет поочередное сравнение выражения со списком константных выражений. При совпадении, выполнение программы начинается с соответствующего оператора. В случае, если совпадений не было, выполняется необязательная ветка default. Важно помнить, что операторы после первого совпадения будут выполняться далее один за другим. Чтобы этого избежать, следует использовать оператор break.

Вывод

В ходе лабораторной работы были освоены оператор выбора switch, а так же операторы case, break, default; циклы for(;;), while(), do while().

Исходный код

menu.c

```
#include "index_first_negative.h"
#include "index_last_negative.h"
#include "multi_between_negative.h"
#include "multi_before_and_after_negative.h"
int main(){
        int B,K,C,L,o,i;
        int N[20];
scanf("%d",&o);
i=0;
while ((getchar())!='\n')
                scanf("%d",&N[i++]);
        switch(o)
        {
                case 0: B=index_first_negative(N,i-1);
                                printf ("%d\n",B);
                        break;

                case 1: K=index_last_negative(N,i-1);
                                printf("%d\n",K);
                        break;

                case 2: C=multi_between_negative(N,i-1);
                       printf("%d\n",C);
                        break;
                case 3: L=multi_before_and_after_negative(N,i-
1);
                       printf("%d\n",L);
                        break;
                default: printf("Данные некорректны");
        }
}
```

index_first_negative.c

```
#include <stdio.h>
                int index_first_negative(int N[],int h){
                int B,i;
                for (i=0;i<=h;i++) {
                                        if (N[i]<0)
                                                {B=i;
```

```
                                        break;
                                        return B;}
                                }
                }
```

index_first_negative.h

```
int index_first_negative(int N[],int );
```

index_last_negative.c

```
#include <stdio.h>

int index_last_negative(int N[],int h){
int m,i;
for (i=h;i>=0;i--) {
                        if (N[i]<0)
                                {m=i;
                        break;
                        return m;}
                }
}
```

index_last_negative.h

```
int index_last_negative(int N[],int h);
```

multi_before_and_after_negative.c

```
#include "index_first_negative.h"
#include "index_last_negative.h"
int multi_before_and_after_negative(int N[],int h)
{
int B,m,i,P;
        B=index_first_negative(N,h);

   m=index_last_negative(N,h);
   P=N[m];
        for (i=0;i<=h;i++){
                if (i<B || i>m)

      {
```

```
          P=P*N[i];}
        }
  return P;
  }
```

multi_before_and_after_negative.h

```
        int multi_before_and_after_negative(int N[],int h);
```

multi_between_negative.c

```
#include<stdio.h>

 #include "index_first_negative.h"
 #include "index_last_negative.h"
 int multi_between_negative(int N[],int h)
 {
 int B,i,m,P;
          B=index_first_negative(N,h);

    m=index_last_negative(N,h);
    P=N[B];
          for (i=0;i<=h;i++){
                  if (i>B && i<m)

        {
          P=P*N[i];}
        }
  return P;
  }
```

multi_between_negative.h

```
  int multi_between_negative(int N[],int h);
```

Makefile

```
menu: menu.o index_first_negative.o index_last_negative.o
multi_between_negative.o multi_before_and_after_negative.o
          gcc menu.o index_first_negative.o index_last_negative.o
  multi_between_negative.o multi_before_and_after_negative.o -o menu
  menu.o: menu.c
          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
multi_between_negative.o: multi_between_negative.c multi_between_negative.h
index_first_negative.c index_first_negative.h index_last_negative.c
index_last_negative.h
	gcc -c multi_between_negative.c
multi_before_and_after_negative.o: multi_before_and_after_negative.c
multi_before_and_after_negative.h index_first_negative.c index_first_negative.h
index_last_negative.c index_last_negative.h
	gcc -c multi_before_and_after_negative.c
clean:
	rm -rf *.o
```