

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

Кушкочева А.О.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

Цель

Освоить операторы выбора switch, а так же операторы case, break, default; циклы 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

○ Функция int abs(int num)

- Прототип: stdlib.h
- Описание:

Функция abs() возвращает модуль целого числа num.

Оператор ветвления switch выполняется следующим образом:

- вычисляется выражение в скобках оператора switch;
- полученное значение сравнивается с метками (константами) в опциях case, сравнение производится до тех пор, пока не будет найдена метка, соответствующая вычисленному значению целочисленного выражения;
- выполняется блок операций соответствующей метки case;
- если соответствующая метка не найдена, то выполнится блок операций, описанный в опции default.

Альтернатива default может отсутствовать, тогда не будет произведено никаких действий.

Вывод

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

Исходный код

index_first_negative.c

```
#include <stdio.h>

#include "index_first_negative.h" int
index_first_negative(int a[], int i){ int
index, first; int flag=1;
for(index=1; (index<i)&&(flag==1); index++)
{
    if(a[index]<0){
first=index-1;          flag=0;
    }
} return
first; }
```

```
index_first_negative.h int  
index_first_negative(int a[], int i);
```

```
index_last_negative.c #include  
  
<stdio.h> #include  
  
"index_last_negative.h" int  
index_last_negative(int a[], int i){  
int last, index; for(index=1;  
index<i+1; index++)  
{  
if(a[index]<0  
) last=index;  
} return last-  
1; } index_last_negative.h int  
index_last_negative(int a[], int  
i);
```

```
multi_between_negative.c #include  
  
<stdio.h>  
  
#include "multi_between_negative.h"  
  
#include "index_first_negative.h"  
  
#include "index_last_negative.h"
```

```

void multi_between_negative(int a[], int i){ int between=1,
index; for(index=index_first_negative(a, i)+1;
index<index_last_negative(a, i); between=a[index]*between;
} printf("%d\n", between);
}

```

multi_between_negative.h

```

void multi_between_negative(int a[], int i);

```

multi_before_and_after_negative.c

```

#include <stdio.h>

#include "multi_before_and_after_negative.h"
#include "index_first_negative.h" #include
"index_last_negative.h" void
multi_before_and_after_negative(int a[], int i){ int index,
multi1=1, multi2=1; for(index=1;
index<index_first_negative(a, i)+1; index++)
multi1=multi1*a[index]; for(index=index_last_negative(a,
i)+1; index<=i; index++) multi2=multi2*a[index]; printf("%d\n",
(multi1*multi2));
}

```

multi_before_and_after_negative.h

```

void multi_before_and_after_negative(int a[], int i); menu.c
#include <stdio.h>

```

```

#include <stdlib.h>

#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 i=0; char
c; int a[100];
while(c!='\n')
{
scanf("%d%c", &a[i], &c);
i++; } i=i-1; switch(a[0]){
    case 0: printf("%d\n",
index_first_negative(a, i)); break;

    case 1:
printf("%d\n", index_last_negative(a,i)); break;

    case 2:
multi_between_negative(a,i); break;

    case 3:
multi_before_and_after_negative(a,i); break;

    default:
printf("Данные некорректны\n");
} return
0;
}

```

Makefile all:

menu clean

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 index_first_negative.h multi_between_negative.h
index_last_negative.h multi_before_and_after_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

multi_between_negative.o: multi_between_negative.c
multi_between_negative.h index_first_negative.h 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.h

index_last_negat\$ gcc -c multi_before_and_after_negative.c

clean:

rm -rf *.o