

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. 7303

Мищенко М.А.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

Оглавление

Цель работы	3
Ход работы	3
Вывод.....	9

Цель работы

В текущей директории создайте 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 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".

Ход работы

1)

`index_first_zero.h`

```
int index_first_zero(int const a[], int size);
```

Объявлена функция, возвращающая **int**, и принимающая массив и его размер. Функция возвращает индекс первого нулевого элемента во входном массиве.

2)

`index_last_zero.h`

```
int index_first_zero(int const a[], int size);
```

Объявлена функция, возвращающая **int**, и принимающая массив и его размер. Функция возвращает индекс последнего нулевого элемента во входном массиве.

3)

sum_between.h

```
int sum_between.c (int const a[], int size);
```

Объявлена функция, возвращающая **int**, и принимающая массив и его размер.

Функция возвращает сумму модулей всех элементов от первого нулевого элемента до последнего нулевого элемента.

4)

sum_before_and_after.h

```
int sum_before_and_after (int const a[], int size);
```

Объявлена функция, возвращающая **int**, и принимающая массив и его размер.

Функция возвращает сумму модулей всех элементов от 0 до первого нулевого элемента и от последнего нулевого элемента до последнего элемента.

5)

index_first_zero.c

```
#include <stdio.h>
```

```
int index_first_zero (int *a,int size) {
```

```
int m;
```

```
for (int i=0;i<size;i++) {
```

```
    if (a[i]==0) {
```

```
        m=i;
```

```
        break;
```

```
    }
```

```
}
```

```
return m;
```

```
}
```

В теле функции выполняется цикл, в котором индекс *i*, проходя по всему массиву останавливается на первом нулевом элементе, присваивает индекс переменной и выводит ее, на чем заканчивается выполнение функции.

6)

index_last_zero.c

```
#include <stdio.h>
int index_last_zero(int *a,int size)
{
    int b,i;
    for (i=size-1;i>=0;i--)
        if (a[i]==0)
        {
            return i;
        }
}
```

В теле функции выполняется цикл, в котором индекс *i*, проходя по всему массиву (начиная с конца массива) останавливается на первом нулевом элементе , и возвращает индекс , на чем заканчивается выполнение функции.

7)

sum_between.c

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "index_last_zero.h"
#include "index_first_zero.h"
int sum_between (int *a,int size)
{
    int n=0,i;
    for (i = index_first_zero(a, size);i < index_last_zero(a, size);i++)
    {
        n+=abs(a[i]);
    }
    return n;
}
```

Подключена библиотека `stdlib.h`, т. к. в дальнейшем в соответствующем файле с кодом будет использоваться функция `abs()`

Подключен соответствующий файлу с кодом заголовочник, и ранее реализованные функции, для нахождения индексов первого и последнего нулевых элементов. В первую очередь нулем инициализируется переменная *n*, затем в цикле индекс проходит от первого нулевого элемента до последнего нулевого на каждой итерации добавляя к *n* модуль *i*-того элемента.

8)

sum_before_and_after.c

```
#include <stdlib.h>
#include <stdio.h>
#include "index_last_zero.h"
#include "index_first_zero.h"
int sum_before_and_after (int a[],int size)
{
    int r=0,n=0,i,sum;
    for (i=0;i<index_first_zero(a,size);i++)
    {
        n+=abs(a[i]);
    }
    for (i=index_last_zero(a, size);i<size;i++)
    {
        r+=abs(a[i]);
    }
    sum=r+n;
    return sum;
}
```

Подключена библиотека `stdlib.h`, т. к. в дальнейшем в соответствующем файле с кодом будет использоваться функция `abs()`

Подключен соответствующий файлу с кодом заголовочник, и ранее реализованные функции, для нахождения индексов первого и последнего нулевых элементов. В первую очередь нулем инициализируются переменные `n` и `r`, `sum` затем в циклах индекс проходит от 0 до первого нулевого элемента на каждой итерации добавляя к `n` модуль `i`-того элемента и от последнего элемента до последнего нулевого элемента на каждой итерации добавляя к `r` модуль `i`-того элемент. Затем переменной `sum` присваивается сумма `r+n` и выводится переменная `sum`.

9)

menu.c

```
#include <stdio.h>
#include <stdlib.h>
#include "sum_before_and_after.h"
#include "sum_between.h"
#include "index_last_zero.h"
#include "index_first_zero.h"
int main () {
    int a[100], number, size = 0, i = 0;
    char c;
    scanf("%d\n", &number);
    while(size < 100) {
        scanf("%d", &a[i++]);
        size++;
        if ((c = getchar()) == '\n') {
            break;
        }
    }

    switch (number) {
        case 0:
            printf("%d\n", index_first_zero(a, size));
            break;
        case 1:
            printf("%d\n", index_last_zero(a, size));
            break;
        case 2:
            printf("%d\n", sum_between(a, size));
            break;
        case 3:
            printf("%d\n", sum_before_and_after(a, size));
            break;
        default:
            printf("Данные некорректны\n");
            break;
    }
    return 0;
}
```

Подключаются все ранее описанные заголовочные файлы и библиотека **stdio.h**, для использования функций **scanf**, **printf**, **getchar**. Инициализируется массив длины 100, и целое **number**. Затем с входного потока считывается значение **number**, и в цикле до тех пор, пока не будет введен символ переноса строки или не будет зафиксирован выход из выделенной области памяти (**size < 100**) будет производиться чтение и поочередное записывание вводимых чисел в массив

Далее оператору **switch** передается `number` и в соответствии с целью работы выводится результат определенной функции. В случае `number` отличающейся от 0, 1, 2, 3 будет выведено сообщение *"Данные некорректны"*

10)

Makefile

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

Для каждого файла с кодом была написана цель с командой, компилирующей его. В зависимостях всех этих целей были указаны соответствующие файл с кодом и заголовочники, которые включались в этот файл.

Заведена цель **menu**, линкующая все необходимые скомпилированные файлы и называющая выходной исполняемый файл *"menu"*.

Цель **clean** для удаления всех объектных файлов.

Вывод

Был создан проект с make-файлом, реализующий требуемый функционал с использованием условных выражений, циклов и оператора **switch**. При компилировании, сборке и выполнении исполняемого файла не возникает ошибок и предупреждений

