

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

Аженилок В.А

Преподаватель

Берленко Т.А

Санкт-Петербург 2017

Цель работы

Познакомиться с операторами case, break и default, с циклами for (;;), while (), do while(), а также с оператором выбора switch .

Задание

В текущей директории необходимо 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.

Нужно 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:

Основные теоретические положения

0 : индекс первого чётного элемента. (index_first_even.c)

1 : индекс последнего нечётного элемента. (index_last_odd.c)

2 : Найти сумму модулей элементов массива, расположенных от первого чётного элемента и до последнего нечётного, включая первый и не включая последний. (sum_between_even_odd.c)

3 : Найти сумму модулей элементов массива, расположенных до первого чётного элемента (не включая элемент) и после последнего нечётного (включая элемент). (sum_before_even_and_after_odd.c)
Иначе необходимо вывести строку "Данные некорректны".

Заголовочные файлы, необходимые для создания проекта:

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

Описание:

Функция вычисляет абсолютную величину (модуль) значения, передаваемого в качестве аргумента через параметр n.

Параметры:

n – целое значение.

Возвращаемое значение:

Модуль числа n.

2. <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

Вывод

В результате работы были освоены оператор выбора switch, циклы for (;), while (), do while(), а также операторы case, break и default, закреплены знания по созданию проектов с Makefile.

Исходный код программы:

1.Makefile

```
bj = index_first_even.o index_last_odd.o sum_between_even_odd.o
sum_before_even_and_after_odd.o menu.o
exe = menu

all: $(obj)
    gcc $(obj) -o $(exe)

menu.o: menu.c index_first_even.h index_last_odd.h sum_between_even_odd.h
sum_before_even_and_after_odd.h
    gcc -c menu.c

index_first_even.o: index_first_even.c
    gcc -c index_first_even.c

index_last_odd.o: index_last_odd.c
    gcc -c index_last_odd.c

sum_between_even_odd.o: sum_between_even_odd.c
    gcc -c sum_between_even_odd.c

sum_before_even_and_after_odd.o: sum_before_even_and_after_odd.c
    gcc -c sum_before_even_and_after_odd.c

clean:
    rm $(obj) $(exe)
```

2.index_first_even.c

```
#include "index_first_even.h"

int index_first_even(int arr[], int arr_size) {
    int i;
    for (i = 0; i < arr_size; ++i)
        if (!(arr[i] & 1))
            return i;
}
```

3.index_first_even.h

```
#ifndef __INDEX_FIRST_EVEN_H__
#define __INDEX_FIRST_EVEN_H__

int index_first_even(int[], int);

#endif
```

4.index_last_odd.c

```
#include "index_last_odd.h"

int index_last_odd(int arr[], int arr_size) {
    int i;
    for (i = arr_size - 1; i >= 0; --i)
        if (arr[i] & 1)
            return i;
}
```

5.index_last_odd.h

```
#ifndef __INDEX_LAST_ODD_H__
#define __INDEX_LAST_ODD_H__

int index_last_odd(int[], int);

#endif
```

6.menu.c

```
#include "index_first_even.h"
#include "index_last_odd.h"
#include "sum_between_even_odd.h"
#include "sum_before_even_and_after_odd.h"

#include <stdio.h>

int arr[100];
int arr_size = 0;
int query;
char c;
```

```

int main() {
    scanf("%d%c", &query, &c);
    while (c != '\n')
        scanf("%d%c", &arr[arr_size++], &c);
    switch(query) {
        case 0:
            printf("%d", index_first_even(arr, arr_size));
            break;
        case 1:
            printf("%d", index_last_odd(arr, arr_size));
            break;
        case 2:
            printf("%d", sum_between_even_odd(arr, arr_size));
            break;
        case 3:
            printf("%d", sum_before_even_and_after_odd(arr, arr_size));
            break;
        default:
            printf("Данные некорректны");
    }
    return 0;
}

```

7.sum_before_even_and_after_odd.c

```

#include "sum_before_even_and_after_odd.h"
#include "index_first_even.h"
#include "index_last_odd.h"

#include <stdlib.h>

int sum_before_even_and_after_odd(int arr[], int arr_size) {
    int a = index_first_even(arr, arr_size);
    int b = index_last_odd(arr, arr_size);
    int sum = 0;
    int i;
    for (i = 0; i < a; ++i)
        sum += abs(arr[i]);
    for (i = b; i < arr_size; ++i)
        sum += abs(arr[i]);
    return sum;
}

```

8. .sum_before_even_and_after_odd.h

```

#ifndef __SUM_BEFORE_EVEN_AND_AFTER_ODD_H__
#define __SUM_BEFORE_EVEN_AND_AFTER_ODD_H__

int sum_before_even_and_after_odd(int[], int);

```

```
#endif
```

9.sum_between_even_odd.c

```
#include "sum_between_even_odd.h"
#include "index_first_even.h"
#include "index_last_odd.h"

#include <stdlib.h>

int sum_between_even_odd(int arr[], int arr_size) {
    int a = index_first_even(arr, arr_size);
    int b = index_last_odd(arr, arr_size);
    int sum = 0;
    int i;
    for (i = a; i < b; ++i)
        sum += abs(arr[i]);
    return sum;
}
```

10. sum_between_even_odd.h

```
#ifndef __SUM_BETWEEN_EVEN_ODD_H__
#define __SUM_BETWEEN_EVEN_ODD_H__

int sum_between_even_odd(int[], int);

#endif
```