

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

Процветкина А. В.

Преподаватель

Берленко Т. А.

Санкт-Петербург

2017

Цель работы

Познакомиться с массивами, оператором выбора switch, циклами for (;), while (), do while (), операторами continue, break и return.

Реализация и сборка функции-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 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".

Основные теоретические положения

Заголовочные файлы, необходимые для создания проекта:

1. <stdlib.h> – содержит прототип функции "int abs(int n);", возвращающей абсолютное значение числа. Используется в определениях функций "int sum_between_even_odd(int [], int);" и "int sum_before_even_and_after_odd(int [], int);". 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 ходе данной лабораторной работы было изучено понятие массивов в Си (синтаксис, использование, расположение в памяти), принципы и основы работы с ними, с оператором выбора `switch`: синтаксисом, использованием, операторами `case`, `break` и `default`. с циклами `for (;;)`, `while ()`, `do while ()`: синтаксисом, использованием, операторами `continue` для перехода к следующей итерации, `break` для выхода из цикла.

Исходный код

- *Файл "index_first_even.h":*

```
int index_first_even (int arr[], int size);
```

- *Файл "index_first_even.c":*

```
#include <stdio.h>

int index_first_even (int arr[], int size){
    int i;
    for (i=0; i<=size; i++)
        if (arr[i]%2==0)
            break;
    return i;
}
```

- *Файл "index_last_odd.h":*

```
int index_last_odd (int arr[], int size);
```

- *Файл "index_last_odd.c":*

```
#include <stdio.h>

int index_last_odd (int arr[], int size){
    int i;
    for (i=size; i>=0; i--)
        if (arr[i]%2==1)
            break;
    return i;
}
```

- *Файл "sum_before_even_and_after_odd.h":*

```
void sum_before_even_and_after_odd (int arr[], int size);
```

- *Файл "sum_before_even_and_after_odd.c":*

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

```

#include <stdlib.h>
#include "index_first_even.h"
#include "index_last_odd.h"
void sum_before_even_and_after_odd (int arr[], int size){
    int i, even, odd;
    even=index_first_even(arr, size);
    odd=index_last_odd(arr, size);
    int sum_even=0, sum_odd=0;
    for (i=0;i<even;i++)
        sum_even += abs(arr[i]);
    for (i=size;odd<=i;i--)
        sum_odd += abs(arr[i]);
    printf("%d\n",sum_odd+sum_even);
}

```

- Файл "sum_between_even_odd.h":

```

void sum_between_even_odd (int arr[], int size);

```

- Файл "sum_between_even_odd.c":

```

#include <stdlib.h>
#include <stdio.h>
#include "index_first_even.h"
#include "index_last_odd.h"
void sum_between_even_odd (int arr[], int size){
    int sum, even, odd;
    even=index_first_even(arr, size);
    odd=index_last_odd(arr, size);
    for (sum=0; even<odd;even++)
        sum+=abs(arr[even]);
    printf("%d\n", sum);
}

```

- *Makefile*

menu: menu.o index_first_even.o index_last_odd.o sum_between_even_odd.o

sum_before_even_and_after_odd.o

gcc menu.o index_first_even.o index_last_odd.o

sum_between_even_odd.o sum_before_even_and_after_odd.o -o menu

menu.o: menu.c

gcc -c menu.c

index_first_even.o: index_first_even.c index_first_even.h

gcc -c index_first_even.c

index_last_odd.o: index_last_odd.c index_last_odd.h

gcc -c index_last_odd.c

sum_between_even_odd.o: sum_between_even_odd.c

sum_between_even_odd.h

gcc -c sum_between_even_odd.c

sum_before_even_and_after_odd.o: sum_before_even_and_after_odd.c

sum_before_even_and_after_odd.h

gcc -c sum_before_even_and_after_odd.c

clean:

rm -rf *.o