МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

ОТЧЕТ

по лабораторной работе №2
по дисциплине «Программирование»
Тема: «Условия, циклы, оператор switch»

Студент гр. 7381 _____ Дорох С.В.

Преподаватель _____ Берленко Т.А.

Санкт-
Петербург
2017

Цель работы:

Создать проект с make-файлом. Главная цель должна приводить к сборке проекта. 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:
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".

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.

Основные теоретические положения:

В ходе лабораторной работы был создан файл menu.c, в котором необходимо вводить с клавиатуры номер операции и массив целых чисел при помощи функции scanf. Потом номер операции поступает в оператор множественного выбора switch и выполняется инструкция, соответствующая введённому номеру операции. При неверном вводе номера операции выводится строка “Данные некорректны”. Всего условий четыре:

0) Индекс первого чётного элемента находится в функции index_first_even;
1) Индекс последнего нечётного элемента находится в функции index_last_odd;
2) Сумма модулей элементов массива, расположенных от первого чётного элемента и до последнего нечётного, включая первый и не включая последний находится в функции sum_between_even_odd;
3) Сумма модулей элементов массива, расположенных до первого чётного элемента (не включая элемент) и после последнего нечётного(включая элемент) находится в функции sum_before_even_and_after_odd.

Для избежания ошибок дублирования кода в функциях sum_between и sum_before вызываются функции first_even и last_odd. При нахождении

модуля суммы использовалась функция abs(val)(val-вещественное значение).

Далее создаётся Makefile, в котором собирается программа и исполняемому файлу даётся имя menu при помощь ключа “-o”.

Вывод:

В данной лабораторной работе изучен оператор множественного выбора switch. Познакомились с операторами case, break, default. Отточены навыки создания make-файла. Познакомились с функцией abs, вычисляющей абсолютное значение и возвращающей модуль значения.

Исходный код проекта:

Файл menu.c:

```
#include <stdio.h>
#include "index_first_even.h"
#include "index_last_odd.h"
#include "sum_between_even_odd.h"
#include "sum_before_even_and_after_odd.h"

int main(){
int arr[100];
int i=0;
char c;
int input=0;
int out=0;

 scanf("%d", &input);
   while ( c != '\n'){
     scanf("%d%c", &arr[i], &c);
     i++;
   }

 switch(input){
   case 0:
     out=index_first_even(arr, i);
     printf("%d\n",out);
     break;
   case 1:
     out=index_last_odd(arr, i);
     printf("%d\n",out);
     break;
```

```
   case 2:
     out=sum_between_even_odd(arr, i);
     printf("%d\n",out);
     break;
   case 3:
     out=sum_before_even_and_after_odd(arr, i);
     printf("%d\n",out);
     break;
    default:
        printf("Данные некорректны\n");
 }
return 0;
}
```

Файл sum_between_even_odd.c:

```
#include <stdlib.h>
#include "index_first_even.h"
#include "index_last_odd.h"

int sum_between_even_odd(int arr[], int k){

int sum=0;
int i=0;

  for (i=(int)index_first_even(arr, k);i<(int)index_last_odd(arr, k);i++){
    sum=sum+ abs(arr[i]);
  }

return sum;

}
```

Файл sum_between_even_odd.h:

```
int sum_between_even_odd(int arr[], int k);
```

Файл sum_before_even_and_after_odd.c:

```
#include <stdlib.h>
#include "index_first_even.h"
#include "index_last_odd.h"
```

```
int sum_before_even_and_after_odd(int arr[], int k){

int sum=0;
int i=0;

 for (i=0;i<index_first_even(arr,k);i++)
   sum = sum + abs(arr[i]);
 for (i=index_last_odd(arr,k);i<k;i++)
   sum = sum + abs(arr[i]);

return sum;
}
```

Файл sum_before_even_and_after_odd.h:

```
int sum_before_even_and_after_odd(int arr[], int k);
```

Файл index_last_odd.c:

```
int index_last_odd(int arr[], int k){
int last=0;
int i=0;

 for (i=0;i<k;i++){
   if (arr[i] %2 !=0)
    last = i;
 }

return last;
}
```

Файл index_last_odd.h:

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

Файл index_first_even.c:

```
#include <stdio.h>

int index_first_even(int arr[], int k){
int first=0;
int i=0;
```

```
 for (i=0;i<k;i++){
   if (arr[i] % 2 ==0)
    {first = i;
    break;
   }
 }

return first;
}
```

Файл index_first_even.h:

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

Makefile:

```
all:menu

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 index_first_even.h index_last_odd.h
sum_before_even_and_after_odd.h sum_between_even_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 index_first_even.h
index_last_odd.h
	gcc -c sum_between_even_odd.c

sum_before_even_and_after_odd.o: sum_before_even_and_after_odd.c
index_first_even.h index_last_odd.h
	gcc -c sum_before_even_and_after_odd.c
```