

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

Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Программирование»

Тема: условия, циклы, оператор switch.

Студент гр. 7381

Преподаватель

Кортев Ю.В.

Берленко Т.А.

Санкт-Петербург

2017

Цель работы.

Освоение практического применения циклических функций и оператора switch.

Создание программы, которая в зависимости от входящего числа выполняет одну из четырех задач:

- 0: Нахождение индекса первого четного числа.
- 1: Нахождение индекса последнего нечетного числа.
- 2: Нахождение суммы модулей элементов массива, расположенных от первого чётного элемента и до последнего нечётного, включая первый и не включая последний.
- 3: Нахождение суммы модулей элементов массива, расположенных до первого чётного элемента (не включая элемент) и после последнего нечётного (включая элемент).

Основные теоретические положения.

Функция нахождения модуля (abs()) находится в библиотеке stdlib.h.

Ход работы.

1. Создаю главный файл menu.c, в нем содержится функция main, на вход в которой подается значение определяющее какую задачу будет решать программа. Также функция main инициализирует и заполняет статический массив a размером 100.
2. Создаю файл index_first_even.c, в нем содержится описание функции ife, аргументом которой является массив a и количество имеющих значение элементов. С помощью цикла for и оператора if функция находит первое четное число и возвращает его индекс.
3. Создаю заголовочный файл index_first_even.h.
4. Создаю файл index_last_odd.c, в нем содержится описание функции ilo, аргументом которой является массив a и количество имеющих значение элементов. Аналогично функции ife, ilo, используя цикл for и оператор if, находит последний нечетный элемент массива.
5. Создаю заголовочный файл index_last_odd.h.
6. Создаю файл sum_between_even_odd.c, в нем содержится описание функции sbeo, аргументом которой является массив a. Используя функции ife и ilo, sbeo находит сумму модулей элементов массива находящихся между первым четным и последним нечетным элементом. Sbeo использует цикл for, который действует пока переменная находится в промежутке между первым четным и последним нечетным элементами массива a.

7. Создаю заголовочный файл sum_between_even_odd.h.
8. Создаю файл sum_before_even_and_after.c, в нем содержится описание функции sbeaa, аргументом которой является массив a. Аналогично функции sbeo, sbeaa находит суммы модулей элементов массива до первого четного и после последнего нечетного элемента, а потом складывает их.
9. Создаю заголовочный файл sum_before_even_and_after.h.
10. Создаю make-файл.

Вывод.

Освоен оператор выбора switch и цикл for.

Исходный код:

-----Index_first_even.c-----

```
#include <stdio.h>
#include "index_first_even.h"
int ife(int a[100], int i)
{
    int n;
    for(n=0;n<i;n++)
    {
        if(a[i]%2==0)
        {
            return i;
        }
    }
}
```

-----Index_first_even.h-----

```
#include <stdio.h>
int ife();
```

-----index_last_odd.c-----

```

#include <stdio.h>
#include "index_last_odd.h"
int ilo(int a[100],int i)
{
int index;
for(index=i;index>0;index--)
{
if(a[index]%2!=0)
{
return index;
}
}
}

```

-----**Index_last_odd.h**-----

```

#include <stdio.h>
int ilo();
sum_between_even_odd.c-----
#include <stdio.h>
#include <stdlib.h>
#include "index_first_even.h"
#include "index_last_odd.h"
#include "sum_between_even_odd.h"
int sbeo(int a[100],int i)
{
int n,sum;
sum=0;
for(n=ife(a);n<ilo(a);n++)
{
sum+=abs(a[n]);
}
return sum;
}

```

-----sum_between_even_and_odd.h-----

```
#include <stdio.h>
#include <stdlib.h>
#include "index_first_even.h"
#include "index_last_odd.h"
int sbeo();
```

-----sum_before_even_and_after.c-----

```
#include <stdio.h>
#include <stdlib.h>
#include "sum_before_even_and_after.h"
#include "index_first_even.h"
#include "index_last_odd.h"
int sbeaa(int a[100],int i)
{
    int n, k ,sum, sum1, sum2;
    sum1=0;
    sum2=0;
    for(n=ife(a)-1; n>=0; n--)
    {
        sum1+=abs(a[n]);
    }

    for(k=ilo(a); k<=i;k++)
    {
        sum2+=abs(a[k]);
    }
    sum=sum1+sum2;
    return sum;
}
```

-----sem_before_even_and_after.h-----

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

```
#include "index_first_even.h"
#include "index_last_odd.h"
int sbeaa();
```

-----**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.h"
int main()
{

    int i;
    int e;
    int k, m, a[100];
    char t;
    scanf("%d",&m);

    for(i=0;i<100;i++)
    {
        scanf("%d%c", &a[i], &t);
        if(t=='\n')
        {
            break;
        }
    }
    switch(m)
    {
        case 0:
            printf("%d\n",ife(a,i));
            break;
```

```

case 1:
printf("%d\n",ilo(a,i));
break;
case 2:
printf("%d\n",sbeo(a,i));
break;
case 3:
printf("%d\n",sbeaa(a,i));
break;
default:
printf("Данные некорректны");
break;
}
}

```

-----Makefile-----

all: menu

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

gcc menu.o index_first_even.o index_last_odd.o sum_between_even_odd.o
sum_before_even_and_after.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
index_last_odd.h index_first_even.h

gcc -c sum_between_even_odd.c

sum_before_even_and_after.o: sum_before_even_and_after.c sum_before_even_and_after.h
index_last_odd.h index_first_even.h

gcc -c sum_before_even_and_after.c