

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

Адамов Я.В.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

Лабораторная работа №1

Условия, циклы, оператор switch

Цель работы: научиться работать с массивами, циклами (while, for), оператором выбора switch.

Задание: Создать проект, который будет состоять из следующих файлов:

- menu.c: реализует главную функцию, принимает целочисленный массив и значение от 0 до 3, в зависимости от которого программа выведет следующее:
- индекс первого нулевого элемента (эта функция реализуется в файле index_first_zero.c).
- индекс последнего нулевого элемента (эта функция реализуется в файле index_last_zero.c).
- сумму модулей элементов массива, расположенных от первого нулевого элемента и до последнего (эта функция реализуется в файле sum_between.c).
- сумму модулей элементов массива, расположенных до первого нулевого элемента и после последнего. (эта функция реализуется в файле sum_before_and_after.c).
- Заголовочных файлов index_first_zero.h
index_last_zero.h sum_between.h sum_before_and_after.h
- Makefile, который содержит инструкцию по сборке программы.

Основные теоретические положения:

Заголовочные файлы стандартной библиотеки языка C, необходимые для выполнения данной лабораторной работы: `stdio.h`, `stdlib.h`.

Библиотека `stdio.h` содержит в себе функции ввода и вывода.

Библиотека `stdlib.h` содержит в себе функцию нахождения модуля `abs()`.

В файле `menu.c` реализована главная функция, которая массив целых чисел от 1 до 100. Числа разделены пробелом. Строка заканчивается символом перевода строки. Так же на вход подаётся число от 0 до 3, в зависимости от значения которого программа выводит следующее (выбор функции происходит с помощью оператора `switch`):

0: индекс первого нулевого элемента. (функция содержится в `index_first_zero.c`)

1: индекс последнего нулевого элемента. (функция содержится в `index_last_zero.c`)

2: сумма модулей элементов массива, расположенных от первого нулевого элемента и до последнего. (функция содержится в `sum_between.c`)

3: сумма модулей элементов массива, расположенных до первого нулевого элемента и после последнего. (функция содержится в `sum_before_and_after.c`)

Иначе выводится строка "Данные некорректны".

Функция `index_first_zero(int[], int)` принимает массив чисел и размер этого массива, с помощью цикла `for()` проходит по всем элементам от начала до первого нулевого и возвращает его индекс.

Функция `index_last_zero(int[], int)` принимает массив чисел и размер этого массива, с помощью цикла `for()` проходит по всем элементам от последнего до первого найденного нулевого и возвращает его индекс.

Функция `sum_between(int[], int)` принимает массив чисел и его размер, с помощью функций `index_first_zero(int[], int)` и `index_last_zero(int[], int)` находит два крайних нулевых элемента и с помощью цикла `for()` суммирует все элементы между, после чего выводит эту сумму.

Функция `sum_before_and_after(int[], int)` принимает массив чисел и его размер, с помощью функций `index_first_zero(int[], int)` и `index_last_zero(int[], int)` находит индексы первого и последнего элемента и с помощью двух циклов `for()` суммирует все элементы стоящие перед первым нулевым и за последним нулевым, после чего выводит эту сумму.

Вывод: были освоены навыки работы с массивами, циклами (`while`, `for`), оператором выбора `switch`.

Исходный код программы:

menu.c:

```
#include <stdio.h>
#include "index_first_zero.h"
#include "index_last_zero.h"
#include "sum_between.h"
#include "sum_before_and_after.h"

int main() {
    int arr[100];
    int arr_size = 0, fun_num, a, b;

    while ((a = getc(stdin)) != '\n') {
        ungetc(a, stdin);
        scanf("%d", &b);
        arr[arr_size++] = b;
    }

    scanf("%d", &fun_num);
    b=0;
    for (int i=0; i<=arr_size; i++){
        if (arr[i]==0)
            b++;
    }
    if (b==0 || ( b<2 && fun_num>1 ))
```

```

        fun_num=4;

switch (fun_num) {
case 0:
    printf("%d", index_first_zero(arr, arr_size));
    break;
case 1:
    printf("%d", index_last_zero(arr, arr_size));
    break;
case 2:
    printf("%d", sum_between(arr, arr_size));
    break;
case 3:
    printf("%d", sum_before_and_after(arr, arr_size));
    break;
default:
    printf("The data is incorrect");
}
return 0
}

```

index_first_zero.h:

```

#pragma once

int index_first_zero(int [], int);

```

index_first_zero.c

```

#include "index_first_zero.h"

int index_first_zero(int arr[], int arr_size) {
    for (int i = 0; i < arr_size; i++)
        if (arr[i] == 0)
            return i;
}

```

index_last_zero.h

```

#pragma once

int index_last_zero(int [], int);

```

index_last_zero.c

```

#include "index_last_zero.h"

```

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

sum_between.h

```
#pragma once
```

```
int sum_between(int [], int);
```

sum_between.c

```
#include <stdlib.h>
#include "sum_between.h"
#include "index_first_zero.h"
#include "index_last_zero.h"

int sum_between(int arr[], int arr_size) {
    int from = index_first_zero(arr, arr_size);
    int to = index_last_zero(arr, arr_size);
    int sum = 0;
    for (int i = from; i < to; i++)
        sum += abs(arr[i]);
    return sum;
}
```

sum_before_and_after.h

```
#pragma once
```

```
int sum_before_and_after(int [], int);
```

sum_before_and_after.c

```
#include <stdlib.h>
#include "sum_before_and_after.h"
#include "index_first_zero.h"
#include "index_last_zero.h"

int sum_before_and_after(int arr[], int arr_size) {
    int to = index_first_zero(arr, arr_size);
    int from = index_last_zero(arr, arr_size);
    int sum = 0;
    for (int i = 0; i < to; i++)
        sum += abs(arr[i]);
}
```

```
    for (int i = from; i < arr_size; i++)  
        sum += abs(arr[i]);  
    return sum;  
}
```

Makefile

```
objects = menu.o index_first_zero.o index_last_zero.o sum_between.o sum_before_and_after.o
```

```
all: $(objects)  
    gcc $(objects) -o manu
```

```
menu.o: index_first_zero.h index_last_zero.h sum_between.h sum_before_and_after.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_between.o: index_first_zero.h index_last_zero.h sum_between.h sum_between.c  
    gcc -c sum_between.c
```

```
sum_before_and_after.o: index_first_zero.h index_last_zero.h sum_before_and_after.h  
sum_before_and_after.c  
    gcc -c sum_before_and_after.c
```

```
clean:  
    rm $(objects) manu
```