

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Программирование»
Тема: Условия, циклы, оператор switch

Студент гр. 0304

Маркуш Н.В.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы:

Изучение процесса сборки программы на языке Си. Применение этих знаний на практике.

Задание:

В текущей директории создайте 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.

Реализуйте 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 нулевого элемента. (`index_first_zero.c`)

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

2 : Найти сумму модулей элементов массива, расположенных от первого нулевого элемента и до последнего. (`sum_between.c`)

3 : Найти сумму модулей элементов массива, расположенных до первого нулевого элемента и после последнего. (`sum_before_and_after.c`)

иначе необходимо вывести строку "Данные некорректны".

Выполнение работы:

В функции `main` файла `main.c` мы объявляем и вводим переменную `int`

act, которая определяет выполняемую операцию. Далее вводятся значения в массив `int input[]` размером в 100 элементов с помощью цикла `for`. С помощью другого цикла `for` переносим значения из массива `int input[]` в массив `int values[]` размером равным количеству введенных элементов.

Затем идёт выполнение выбранной операции. Функции, соответствующие каждой из операций описаны в отдельных файлах (`index_first_zero.c`, `index_last_zero.c`, `sum_between.c`, `sum_before_and_after.c`). К файлу `menu.c` все функции подключены с помощью соответствующих `.h` файлов (`index_first_zero.h`, `index_last_zero.h`, `sum_between.h`, `sum_before_and_after.h`). Так же для выполнения функций `sum_between.h` и `sum_before_and_after.h` необходимо подключить к ним функции `index_first_zero.h`, `index_last_zero.h` и библиотеку `stdlib.h`.

Файлы линкуются с помощью `Makefile`.

Тесты:

№ теста	Вход	Выход	Комментарий
1	1 2 3 4 0 2 3 4 0 2	7	
2	3 2 -3 4 0 2 -2 8 4 0 2 -3 4 8 0 2	11	
3	0 2 3 4 0 -2 4 -4 2 4 7 6 0 0 3	3	

Вывод:

Был изучен и применен на практике процесс сборки программы на языке Си. Задание из лабораторной работы №1 было разбито на главный файл `menu.c`, файлы-функции, файлы-функции были подключены к `menu.c` с помощью `.h` файлов. Затем все это было собрано с помощью `Makefile`.

Исходный код программы:

main.c:

```
#include <stdio.h>
#include <stdlib.h>
#include "index_first_zero.h"
#include "index_last_zero.h"
#include "sum_before_and_after.h"
#include "sum_before_and_after.h"
#define MAX_SIZE 100

int main(){

    int act;

    scanf("%d", &act);

    char space;

    int input[MAX_SIZE];

    int size = 0;

    for(int i = 0; i < MAX_SIZE; i++){

        scanf("%d%c", &input[i], &space);

        size++;

        if(space == '\n'){

            break;

        }

    }

    int values[size];

    for(int i = 0; i < size; i++){

        values[i] = input[i];

    }

    switch ( act ){

        case 0:

            printf("%d\n", index_first_zero(values, act, size));

            break;
```

```

        case 1:
            printf("%d\n", index_last_zero(values, act, size));
            break;

        case 2:
            printf("%d\n", sum_between(values, act, size));
            break;

        case 3:
            printf("%d\n", sum_before_and_after(values, act, size));
            break;

        default:
            printf("Данные некорректны\n");
    }

    return 0;
}

```

index_first_zero.c:

```

#include "index_first_zero.h"

int index_first_zero(int values[], int act, int size){
    int zero_first = 0;
    int first = 0;

    for(int i = 0; i < size; i++){
        if(values[i] == 0){
            first = i;
            zero_first = 1;
            break;
        }
    }
}

```

```
    return first;
}
```

index_last_zero.c:

```
#include "index_last_zero.h"

int index_last_zero(int values[], int act, int size){
    int zero_last = 0;
    int last = 0;

    for(int i = (size - 1); i >= 0; i--){
        if(values[i] == 0){
            last = i;
            zero_last = 1;
            break;
        }
    }

    return last;
}
```

sum_between.c:

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

int sum_between(int values[], int act, int size){
    int first = index_first_zero(values, act, size);
    int last = index_last_zero(values, act, size);
    int sum = 0;

    for(int i = first; i < last; i++){
        sum += abs(values[i]);
    }
}
```

```
return sum;
```

```
}
```

sum_before_and_after.c:

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

```
int sum_before_and_after(int values[], int act, int size){
    int all_sum = 0;

    for(int i = 0; i < size; i++){
        all_sum += abs(values[i]);
    }

    int sum;

    sum = all_sum - sum_between(values, act, size);

    return sum;
}
```

index_first_zero.h:

```
int index_first_zero(int values[], int act, int size);
```

index_last_zero.h:

```
int index_last_zero(int values[], int act, int size);
```

sum_between.h:

```
int index_last_zero(int values[], int act, int size);
```

sum_before_and_after.h:

```
int sum_between(int values[], int act, int size);
```

Makefile:

all: lab2

```
lab2: menu.o index_first_zero.o index_last_zero.o sum_between.o  
sum_before_and_after.o  
    gcc menu.o index_first_zero.o index_last_zero.o sum_between.o  
sum_before_and_after.o -o menu
```

```
menu.o: menu.c sum_before_and_after.h sum_between.h index_last_zero.h  
index_first_zero.h  
    gcc -c -std=c99 menu.c
```

```
index_first_zero.o: index_first_zero.c index_first_zero.h  
    gcc -c -std=c99 index_first_zero.c
```

```
index_last_zero.o: index_last_zero.c index_last_zero.h  
    gcc -c -std=c99 index_last_zero.c
```

```
sum_between.o: sum_between.c sum_between.h index_last_zero.h  
index_first_zero.h  
    gcc -c -std=c99 sum_between.c
```

```
sum_before_and_after.o: sum_before_and_after.c sum_before_and_after.h  
sum_between.h index_last_zero.h index_first_zero.h  
    gcc -c -std=c99 sum_before_and_after.c
```

```
clean:  
    rm -rf *.o lab2
```