

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: Сборка программ в Си.

Студент: Кардаш Я. Е.

Группа: 1304

Преподаватель: Чайка К. В.

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 файле `menu.c` в функции `main` с помощью переменной `int numb_les` происходит выбор операции. Затем в цикле заполняется массив `int arr[]`. Реализована возможность заполнения менее чем 100 эл-тов путем нажатия `enter` (с помощью переменной `char c` и условия)

Далее происходит одна из четырех операций. Каждая функция (соответствующая операция) написана в отдельном файле. Задача каждой функции описана в условии. К файлу menu.c подключены все четыре файла с заголовками функций (.h файлы). К файлам функций (.c) также подключены .h соответствующие файлы. Кроме того к sum_between.c и sum_before_and_after.c подключены также на index_first_zero.h и index_last_zero.h, так как это необходимо для их работы.

Все файлы линкуются с помощью Makefile.

Тесты:

№ задания	Вход	Выход	Комментарий
0	0 1 4 5 0 6 0	3	Индекс 1-ого нулевого эл-та =3
1	1 1 4 5 0 6 0	5	Индекс последнего нулевого эл-та =5
2	2 0 4 -5 1 0	10	Сумма модулей эл-тов массива между первым и последним эл-том =0
3	3 2 6 0 6 1 0 -5	13	Сумма модулей эл-тов массива за пределами промежутка между первым и последним эл-том =0

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:

Menu.c:

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

```

#include <stdlib.h>
#define N 100
#include "index_first_zero.h"
#include "index_last_zero.h"
#include "sum_between.h"
#include "sum_before_and_after.h"
int main()
{
    int numb_les;

    /*printf("Введите номер задания и заполните
массив");*/

    scanf("%d ", &numb_les);
    int arr[N];
    int n=0;
    char c;
    while (n<N) {
        scanf("%d%c", &arr[n], &c);
        n++;
        if (c == '\n') {
            break;}
    }

    switch (numb_les) {
        case 0: {printf("%d\n", index_first_zero(arr, n));
            break;}
        case 1: {printf("%d\n", index_last_zero(arr, n));
            break;}
        case 2: {printf("%d\n", sum_between(arr, n));
            break;}
    }
}

```

```

        case 3:
{printf("%d\n",sum_before_and_after(arr,n));
        break;}

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

return 0;
}

```

```

Index_first_zero.c:
#include "index_first_zero.h"
int index_first_zero(int arr[],int n){
    int index=0;
    for (int i=0;i<n;i++){
        if (arr[i]==0){
            index=i;
            break;}
    }
return index;}

```

```

Index_last_zero.c:
#include "index_last_zero.h"
int index_last_zero(int arr[],int n){
    int index=0;
    for (int i=n-1;i>=0;i--){
        if (arr[i]==0){
            index=i;
            break;}
    }
}

```

```

        return index;
    }

Sum_between.c:
#include "sum_between.h"
#include "index_first_zero.h"
#include "index_last_zero.h"
#include <stdlib.h>

int sum_between(int arr[],int n){
    int begin = index_first_zero(arr,n);
    int end = index_last_zero(arr,n);
    int sum =0;
    for (int i = begin+1;i<end;i++){
        sum+= abs(arr[i]);
    }
    return sum;
}

Sum_before_and_after.c:
#include "sum_before_and_after.h"
#include "index_first_zero.h"
#include "index_last_zero.h"
#include <stdlib.h>

int sum_before_and_after(int arr[],int n){
    int begin = index_first_zero(arr,n);
    int end = index_last_zero(arr,n);
    int sum =0;
    for (int i = 0;i<begin;i++){
        sum+= abs(arr[i]);
    }
    for (int i =end+1;i<n;i++){

```

```

        sum+= abs(arr[i]);
    }
    return sum;
}

Index_first_zero.h:
int index_first_zero(int arr[],int n);
Index_last_zero.h:
int index_last_zero(int arr[],int n);

Sum_before_and_after.h:
int sum_before_and_after(int arr[],int n);

Sum_between.h:
int sum_between(int arr[],int n);

Makefile:
all: 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
    gcc -c menu.c
index_first_zero.o: index_first_zero.c
    gcc -c index_first_zero.c
index_last_zero.o: index_last_zero.c
    gcc -c index_last_zero.c
sum_between.o: sum_between.c
    gcc -c sum_between.c
sum_before_and_after.o: sum_before_and_after.c
    gcc -c sum_before_and_after.c

```

