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

ОТЧЕТ

по лабораторной работе №1
по дисциплине «Программирование»
Тема “Сборка. Make”

Студентка гр. 1384 _____ Логинова А. Ю.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург
2021

Цель работы.

Применить основные конструкции C в разработке, научиться создавать Makefile.

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:

1. Индекс первого отрицательного элемента. (index_first_negative.c)
2. Индекс последнего отрицательного элемента. (index_last_negative.c)
3. Найти сумму модулей элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент). (sum_between_negative.c)
4. Найти сумму модулей элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент). (sum_before_and_after_negative.c)

иначе необходимо вывести строку "Данные некорректны".

Выполнение работы.

На вход программе подается целое значение, которое инициализируется в переменной *task* и будет указывать на дальнейший вывод программы. Далее, через пробел, программа принимает на вход массив целых чисел, который

инициализируется в переменной *user_input,* для которой уже был определен максимальный размер 100, исходя из условий. Во время считывания этого массива также определяется переменная *size*, которая указывает на количество переданных значений. Считывание прекращается, как только программа принимает на вход *\n,* с помощью функции *fgetc(),* считывающей символ из стандартного потока ввода.

В зависимости от значения *task*, в консоль программы выводится одно из 5 значений.

Если *task* = 0, программа вызывает функцию *index_first_negative,* в которую передает массив целых чисел *user_input* и размер, определенных пользователем значений, в переменной *size*. В функции определен цикл, который ищет первое отрицательное значение в массиве *user_input* и возвращает его индекс.

Если *task* = 1, программа вызывает функцию *index_last_negative,* в которую передает массив целых чисел *user_input* и *size*. В функции определен цикл, который ищет последнее отрицательное значение в массиве *user_input* и возвращает его индекс.

Если *task* = 2, программа вызывает функцию *sum_between_negative,* в которую передает массив целых чисел *user_input, size,* а также индексы первого и последнего отрицательных значений *user_input*. В функции определен цикл, который суммирует модули значений массива *user_input* между первым и последним отрицательным значением в переменную *sum* и возвращает ее.

Если *task* = 3, программа вызывает функцию *sum_before_and_after_negative,* в которую передает массив целых чисел *user_input, size,* индексы первого и последнего отрицательных значений *user_input*. В функции определен цикл, который суммирует модули значений массива *user_input* до первого отрицательного значения и после последнего. Сумма значений лежит в переменной sum, которую и возвращает функция.

Если task не равен ни одному из вышеперечисленных значений, программа выводит в консоль «Данные некорректны».

Был создан *Makefile,* в котором описана последовательность сборки данной программы. Включения и функции были распределены по исходным файлам соответственно их названиям: *menu.c, index_first_negative.c, index_last_negative.c, sum_between_negative.c, sum_before_and_after_negative.c.* Были созданы заголовочные файлы для каждого исходного файла, в которые были помещены прототипы функций: *index_first_negative.h, index_last_negative.h, sum_between_negative.h, sum_before_and_after_negative.h.*

Код программы см. в приложении А.

Тестирование.

№ п/п	Входные данные	Выходные данные	Комментарии
1	0 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	3	Программа работает корректно.
2	5 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	Данные некорректны	Программа должна получить запрос от 0 до 3.

3	1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 11 1 1 1 2 -5 4 3 9 1	95	Программа проигнорирует все значения, которые не поместились в массив и выведет индекс не последнего отрицательного значения, а последнего отрицательного значения в самом массиве.
4	3 1 1 1 1 1 1 1 1 1	-1	Если все значения положительные, функции не смогут работать корректно и в качестве несуществующего индекса/суммы вернут -1.

Выводы.

Были изучены возможности для автоматической сборки программы с помощью make.

ПРИЛОЖЕНИЕ А

Исходный код программы

Название файла: menu.c

```
#include <stdio.h>
#include "index_first_negative.h"
#include "index_last_negative.h"
#include "sum_between_negative.h"
#include "sum_before_and_after_negative.h"

#define ARR_SIZE 100

int main(){
    int task, user_input[ARR_SIZE] = {0}, size = 0;
    scanf("%d", &task);
    do{
        scanf("%d", &user_input[size]);
        ++size;
    } while (fgetc(stdin) != '\n' && size < ARR_SIZE);

    switch(task){
        case 0:
            printf("%d\n", index_first_negative(user_input, size));
            break;
        case 1:
            printf("%d\n", index_last_negative(user_input, size));
            break;
        case 2:
            printf("%d\n", sum_between_negative(user_input, size));
            break;
        case 3:
            printf("%d\n", sum_before_and_after_negative(user_input,
size));
            break;
        default:
            printf("Данные некорректны");
            break;
    }
    return 0;
}
```

Название файла: index_first_negative.c

```
#include "index_first_negative.h"

int index_first_negative(int arr[], int size){
    int i;
    for(i = 0; i < size; ++i){
        if (arr[i] < 0)
            return i;
    }
    return -1;
}
```

Название файла: index_last_negative.c

```
#include "index_last_negative.h"

int index_last_negative(int arr[], int size){
    int i;
    for(i = size - 1; i >= 0; --i){
        if (arr[i] < 0)
            return i;
    }
    return -1;
}
```

Название файла: sum_between_negative.c

```
#include <stdlib.h>
#include "index_first_negative.h"
#include "index_last_negative.h"
#include "sum_between_negative.h"

int sum_between_negative(int arr[], int size){
    int i, sum = 0,
        first_index = index_first_negative(arr, size),
        last_index = index_last_negative(arr, size);

    if (first_index == -1 || last_index == -1)
        return -1;
    for(i = first_index; i < last_index; ++i)
        sum += abs(arr[i]);

    return sum;
}
```

Название файла: sum_before_and_after_negative.c

```
#include <stdlib.h>
#include "sum_before_and_after_negative.h"
#include "index_first_negative.h"
#include "index_last_negative.h"

int sum_before_and_after_negative(int arr[], int size){
    int i, sum = 0,
        first_index = index_first_negative(arr, size),
        last_index = index_last_negative(arr, size);

    if (first_index == -1 || last_index == -1)
        return -1;
    for(i = 0; i < first_index; ++i)
        sum += abs(arr[i]);
    for(i = last_index; i < size; ++i)
        sum += abs(arr[i]);

    return sum;
}
```

Название файла: sum_before_and_after_negative.h

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

Название файла: sum_between_negative.h

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

Название файла: index_first_negative.h

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

Название файла: index_last_negative.h

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