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 С 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 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.с

#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.с

#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.с

#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.с

#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.с

#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);