

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

Студентка гр. 0382

Охотникова Г.С.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2020

### **Цель работы.**

Изучение сборки проекта с помощью make-файла.

### **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 отрицательного элемента. (*index\_first\_negative.c*)

1 : индекс последнего отрицательного элемента. (*index\_last\_negative.c*)

2 : Найти сумму модулей элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент). (*sum\_between\_negative.c*)

3 : 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".

### **Основные теоретические положения.**

Компиляция - процесс преобразования программы с исходного языка высокого уровня в эквивалентную программу на языке более низкого уровня (в частности, машинном языке).

Компилятор - программа, которая осуществляет компиляцию.

Линковщик (компоновщик) - он принимает на вход один или несколько объектных файлов и собирает по ним исполняемый модуль. Работа компоновщика заключается в том, чтобы в каждом модуле определить и связать ссылки на неопределённые имена.

Сборка проекта - это процесс получения исполняемого файла из исходного кода.

Makefile - список инструкций для утилиты make, которая позволяет собирать проект сразу целиком.

Структура make-файла:

Любой make-файл состоит из:

- 1) списка целей
- 2) зависимостей этих целей
- 3) команд, которые требуется выполнить, чтобы достичь эту цель

### **Выполнение работы.**

*length* — счетчик длины массива, передается в функции.

*a[C]* — массив, который подается на вход.

*a\_size* — размер массива.

*sym* — символ, обозначающий пробел.

*res1, res2, res3, res4* — переменные, которым присваивается значение, которое возвращают функции.

*n* — переменная, обозначающая выбор команды (0. 1. 2 или 3).

*#define C 100* — макрос, задающий максимальный размер массива.

Функция *index\_first\_negative(int a[C], int lenght)* принимает на вход массив *a*, максимальный размер которого *C*, и длину введенного массива *length*. Переменная *r = -1* является обозначением индекса первого отрицательного элемента, также она нужна для сравнения индекса элемента в массиве, так как нужно найти индекс первого отрицательного элемента, в каждой итерации цикла *for* идет проверка значения *r*. Если *r = -1*, то цикл продолжается. В обратном случае, нужный индекс найден. Функция возвращает найденное значение. Объявление функции находится в файле *index\_first\_negative.h*. Определение функции находится в файле *index\_first\_negative.c*.

Функция *index\_last\_negative(int a[C], int lenght)* принимает на вход массив *a*, максимальный размер которого *C*, и длину введенного массива *length*. Переменная *i1* будет обозначать индекс последнего отрицательного элемента. В цикле *for* идет проверка каждого элемента массива на отрицательность. Таким образом, функция вернет индекс последнего отрицательного элемента в массиве. Объявление функции находится в файле *index\_last\_negative.h*. Определение функции находится в файле *index\_last\_negative.c*.

Функция *sum\_between\_negative(int a[C], int lenght)* принимает на вход массив *a*, максимальный размер которого *C*, и длину введенного массива *length*. Переменная *sum* — это искомая сумма. Переменные *i1* и *i2* обозначают индексы первого отрицательного элемента и последнего соответственно. Для получения этих значений используется вызов первых двух функций. Затем с помощью цикла *for* считается сумма модулей(функция *abs*) элементов от первого отрицательного элемента до последнего. Функция возвращает значение этой

суммы. Объявление функции находится в файле *sum\_between\_negative.h*. Определение функции находится в файле *sum\_between\_negative.c*.

Функция *sum\_before\_and\_after\_negative(int a[C], int lenght)* принимает на вход массив *a*, максимальный размер которого *C*, и длину введенного массива *length*. Переменная *sum1* — это сумма модулей элементов до первого отрицательного, а переменная *sum2* — это сумма модулей элементов после последнего отрицательного. Переменные *i1* и *i2* обозначают индексы первого отрицательного элемента и последнего соответственно. Для получения этих значений используется вызов первых двух функций. Функция разбита на два цикла *for*, в которых отдельно считаются суммы модулей до первого отрицательного элемента и после последнего. Функция возвращает сумму переменных *sum1* и *sum2*. Объявление функции находится в файле *sum\_before\_and\_after\_negative.h*. Определение функции находится в файле *sum\_before\_and\_after\_negative.c*.

В функции *main* сначала на вход поступает выбранная команда (0, 1, 2 или 3), затем массив. В зависимости от выбранной команды оператор множественного выбора вызывает соответствующую команде функцию и выводит результат. Если выбранной команды не существует, выводится сообщение «Данные некорректны». Определение функции находится в файле *menu.c*.

В директории со всеми вышеописанными файлами создается *Makefile*, в котором прописываются команды для компиляции программы. С помощью утилиты *make* собирается программа.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

| № п/п | Входные данные   | Выходные данные    | Комментарии  |
|-------|--|--------------------|--|
| 1.    | 0 1 16 2 -18 -22 15 -3 13 0<br>-6 1 9 24 1 -18 15 28 20<br>-17 16 -11    | 3                  | Индекс первого отрицательного элемента равен 3.  |
| 2.    | 1 1 16 2 -18 -22 15 -3 13 0<br>-6 1 9 24 1 -18 15 28 2<br>-17 16 -11     | 20                 | Индекс последнего отрицательного элемента равен 20.  |
| 3.    | 2 1 16 2 -18 -22 15 -3 13 0<br>-6 1 9 24 1 -18 15 28 20 -<br>17 16 -11   | 226                | Сумма модулей элементов между первым отрицательным элементом(включая) и последним отрицательным(не включая) равна 226. |
| 4.    | 3 1 16 2 -18 -22<br>15 -3 13 0 -6 1 9<br>24 1 -18 15 28 20<br>-17 16 -11 | 30                 | Сумма модулей элементов до первого отрицательного(не включая) и после последнего отрицательного(включая) равна 30.     |
| 5.    | 5 4 5 7 9 -2 4   | Данные некорректны | Введена несуществующая команда. Выведено сообщение об ошибке.  |
| 6.    | 2 23 3 4 5 0 0 -8<br>-11 4 5 78 34 0 0                                   | 8                  | Сумма модулей элементов между первым отрицательным элементом(включая) и последним отрицательным(не включая) равна 8.   |

## **Выводы.**

Была изучена сборка программ в Си с помощью *make-файла*.

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. Для выбора команды используется оператор *switch*. Для обработки команд в функциях используются условные операторы *if* и *while*, а также цикл *for*. Программа собирается из нескольких файлов и компилируется с помощью утилиты *make*.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: index\_first\_negative.h

```
##define C 100  
int index_first_negative(int a[C], int lenght);
```

Название файла: index\_last\_negative.h

```
#define C 100  
int index_last_negative(int a[C], int lenght);
```

Название файла: sum\_between\_negative.h

```
#define C 100  
int sum_between_negative(int a[C], int lenght);
```

Название файла: sum\_before\_and\_after\_negative.h

```
#define C 100  
int sum_before_and_after_negative(int a[C], int lenght);
```

Название файла: index\_first\_negative.c

```
#include <stdio.h>  
#include "index_first_negative.h"  
#define C 100  
  
int index_first_negative(int a[C], int lenght) {  
    int r = -1;  
    for (int i = 0; i < lenght; i++) {  
        if (a[i] < 0 && r == -1) {  
            r = i;  
        }  
    }  
    return r;  
}
```

Название файла: index\_last\_negative.c

```
#include <stdio.h>  
#include "index_last_negative.h"  
#define C 100
```



```

int index_last_negative(int a[C], int lenght) {
    int i1;
    for (int i = 0; i < lenght; i++) {
        if (a[i] < 0) {
            i1 = i;
        }
    }
    return i1;
}

```

Название файла: sum\_between\_negative.c

```

#include <stdio.h>
#include <stdlib.h>
#include "sum_between_negative.h"
#include "index_first_negative.h"
#include "index_last_negative.h"
#define C 100

int sum_between_negative(int a[C], int lenght) {
    int sum = 0;
    int i1, i2, i;
    i1 = index_first_negative(a, lenght);
    i2 = index_last_negative(a, lenght);
    for (i = i1; i < i2; i++) {
        sum = sum + abs(a[i]);
    }
    return sum;
}

```

Название файла: sum\_before\_and\_after\_negative.c

```

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

```

```

#define C 100

int sum_before_and_after_negative(int a[C], int lenght) {
    int sum1 = 0, sum2 = 0, sum = 0;
    int i;
    int i1, i2;
    i1 = index_first_negative(a, lenght);
    i2 = index_last_negative(a, lenght);
    for (i = 0; i < i1; i++) {
        sum1 = sum1 + abs(a[i]);
    }
    for (i = i2; i < lenght; i++) {
        sum2 = sum2 + abs(a[i]);
    }
    sum = sum1 + sum2;
    return sum;
}

```

Название файла: menu.c

```

#include <stdio.h>
#include <stdlib.h>
#include "index_first_negative.h"
#include "index_last_negative.h"
#include "sum_between_negative.h"
#include "sum_before_and_after_negative.h"
#define C 100

int main()
{
    int lenght = 0;
    int a[C];
    int a_size = 0;
    char sym = ' ';
    int res1, res2, res3, res4, n;
    scanf("%d", &n);
}

```

```

while (a_size < C && sym == ' ') {
    scanf("%d%c", &a[a_size++], &sym);
    lenght++;
}

switch (n) {

case 0:
    res1 = index_first_negative(a, lenght);
    printf("%d", res1);

    break;

case 1:
    res2 = index_last_negative(a, lenght);
    printf("%d", res2);
    break;

case 2:
    res3 = sum_between_negative(a, lenght);
    printf("%d", res3);
    break;

case 3:
    res4 = sum_before_and_after_negative(a, lenght);
    printf("%d", res4);

    break;

default:
    printf("Данные некорректны");
    return 0;
}

return 0;
}

```

## Название файла: Makefile

```
all: menu.o index_first_negative.o index_last_negative.o sum_be-
tween_negative.o sum_before_and_after_negative.o

gcc menu.o index_first_negative.o index_last_negative.o
sum_between_negative.o sum_before_and_after_negative.o -o menu

index_first_negative.o: index_first_negative.c
gcc -c index_first_negative.c -std=c99

index_last_negative.o: index_last_negative.c
gcc -c index_last_negative.c -std=c99

sum_between_negative.o: sum_between_negative.c index_first_nega-
tive.h index_last_negative.h
gcc -c sum_between_negative.c -std=c99

sum_before_and_after_negative.o: sum_before_and_after_negative.c
index_first_negative.h index_last_negative.h
gcc -c sum_before_and_after_negative.c -std=c99

menu.o: menu.c index_first_negative.h index_last_negative.h
sum_between_negative.h sum_before_and_after_negative.h

gcc -c menu.c -std=c99
```