**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: Сборка программ в Си**

Студент гр. 1304 _____ Стародубов М.В.

Преподаватель _____ Чайка К.В.

Санкт-Петербург

**2021**

**Цель работы.**

Научиться создавать программу, состоящую из нескольких файлов исходного кода. Научиться создавать и использовать заголовочные файлы, *make*-файлы, раздельно компилировать файлы исходного кода, а также компановать объектные файлы в программу.

**Задание.**

Вариант – 6.

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.**

В заголовочных файлах, имеющих расширение *.h*, находятся директивы

препроцессора, а также определения функций. Для вызова зголовочного файла, в файле исходного коде необходимо прописать

```
#include «название_заголовочного_файла.h»
```

Для раздельной компиляции одного из файлов программы необходимо вызвать компилятор *gcc* с флагом *-c*, и передать ему на вход название файла, который необходимо скомпилировать. Это приведет к созданию объектного файла, имеющего расширение *.o*.

Для компановки программы необходимо передать компилятору gcc названия необходимых для компоновки файлов (объектных файлов или файлов исходного кода). Для того, чтобы задать имя исполнчемому файлу, нужно после перечисления всех необходимых файлов прописать флаг *-o* и написать нужное имя.

*Makefile* необходим для быстрой компиляции программы, состоящей из нескольких файлов. В *make*-файле цели прописываются последовательно следующим образом:

```
<название цели>: <файлы, необходимые для исполнения цели>
        <команда для исполнения цели>
```

**Выполнение работы.**

1. Заголовочные файлы.

В заголовочном файле *index_first_negative.h* назодится описание функции *index_first_negative*.

В зазоловочном файле *index_last_negative.h* находится описание функции index_last_negative.

В заголовочном файле *sum_between_negative.h* подклячены заголовочные файлы *index_first_negative.h*, *index_last_negative.h*, заголовочный файл стандартной библиотеки *stdlib.h*, а также находаится описание функции *sum_between_negative*.

В заголовочном файле *sum_before_and_after_negative.h* подклячены

заголовочные файлы *index_first_negative.h*, *index_last_negative.h*, заголовочный файл стандартной библиотеки *stdlib.h*, а также находаится описание функции *sum_before_and_after_negative*.

2. Файл *menu.c*.

В файле подключены заголовочные файлы *index_first_negative.h*, *index_last_negative.h*, *sum_between_negative.h* , *sum_before_and_after_negative.h*, *stdio.h*, а также определено значение *DEFAULT_SIZE* – максимаьлный размер массива чисел, идущих на вход программе. Далее находиться определение функции main. В переменную *function_index* записывается первая цифра из входных данных, а в переменную *last_character* записывается символ, идущий после введенной цифры. Далее создается целочисленный массив *numbers*, в который будет записан массив целых чисел, не привышающий по размерам значение *DEFAULT_SIZE*. В данный массив будет записан поступающий далее на вход массив целых чисел. В переменную *array_size* будет записан размер данного массива. В цикле *for* в массив *numbers* последовательно записываются значения поступающие на вход программе. С каждой новой итерацией цикла значение *array_size* увеличивается на единицу. Если ввод очередного числа оканчивается символом перевода стрки, то выполнение цикла прекращается. Далее с помощью условного оператора *switch* и переменной *function_index* согласкно заданию определяется, использование какой функции запрашивает пользователь. В случае, если введенный индекс не соответствует ни одному из возможных, то программа выведет на экран сообщение „Данные некорректны“ и завершит работу. В остальных случаях будет вызвана соответствующая индексу функция, а после ее завершения на экран будет выведено полученное от функции значение и программа завершит работу.

3. Файл *index_first_negative.c*.

В файле подключен заголовочный файл *index_first_negative.h*. Далее

находится определение функции *index_first_negative*. Функция принимает на вход массив целых чисел, а также размер данного массива. Тело данной функции состоит из одного цикла, который перебирает начиная с 0 индексы полученного на вход массива, и в случае, если соответствующий данному индексу элемент массива - это отрицательное число, то функция возвращает данный индекс.

4. Файл *index_last_negative.c*.

В файле подключен заголовочный файл *index_last_negative.h*. Далее находится определение функции *index_last_negative*. Функция принимает на вход массив целых чисел, а также размер данного массива. Тело данной функции состоит из одного цикла, который перебирает индексы начиная с последнего, в порядке убывания. В случае, если соответствующий данному индексу элемент массива отрицательный, то программа возвращает данный индекс.

5. Файл *sum_between_negative.c*.

В файле подключен заголовочный файл *sum_between_negative.h*. Далее находится определение функции *sum_between_negative*. Функция принимает на вход массив целых чисел, а также размер данного массива. В переменную *sum* будет записана сумма чисел, расположенных между первым и последним отрицательными числами в массиве. В цикле идет перебор индексов элементов, начиная с индекса первого отрицательного элемента включительно, заканчивая индексом последнего отрицательного элемента невключительно. Данные индексы функция получает, используя функции *index_first_negative* и *index_last_negative*. В теле цикла в переменную *sum* мы суммируем модули всех элементов, соответствующих перебираемым индексам. Функция возвращает полученную сумму.

6. Файл *sum_before_and_after_negative.c*.

В файле подключен заголовочный файл *sum_before_and_after_negative.h*. Далее находится определение функции *sum_before_and_after_negative*. Функция принимает на вход массив целых чисел, а также размер данного массива. В переменной *sum* будет записана сумма элементов массива. В теле функции присутствует два цикла. Первый цикл с помощью функции *index_first_negative* получает индекс первого отрицательного элемента, и перебирает все индексы, начиная с нуля, до индекса первого отрицательного элемента невключительно. В теле цикла в переменной *sum* суммируются модули элементов массива, соответствующие данным индексам. Во втором цикле с помощью функции *index_last_negative* мы получаем индекс последнего отрицательного элемента, и начиная с него перебираем индексы до конца массива. В теле цикла в переменную *sum* суммируются модули элементов массива, соответствующих перебираемым в этом цикле индексам. Функция возвращает записанную в переменной *sum* сумму.

7. Файл *Makefile*.

Цель *all* приводит к компановке всех объектных файлов. Для исполенния цели необходимы файлы *menu.o, index_first_negative.o, index_last_negative.o, sum_between_negative.o, sum_before_and_after_negative.o, index_first_negative.h, index_last_negative.h, sum_between_negative.h, sum_before_and_after_negative.h*.

Цель *menu.o* приводит к компилированию файла *menu.c*. Для исполенения цели необходимы файлы *menu.c, index_first_negative.h, index_last_negative.h, sum_between_negative.h, sum_before_and_after_negative.h*.

Цель *index_first_negative.o* приводит к компилированию файла *index_first_negative.c*. Для исполнения цели необходимы файлы *index_first_negative.c*, *index_first_negative.h*.

Цель *index_last_negative.o* приводит к компилированию файла *index_last_negative.c*. Для исполнения цели необходимы файлы

*index_last_negative.c, index_last_negative.h*.

Цель *sum_between_negative.o* приводит к компилированию файла *sum_between_negative.c*. Для исполнения цели необходимы файлы *sum_between_negative.c, sum_between_negative.h, index_first_negative.h, index_last_negative.h*.

Цель *sum_before_and_after_negative.o* приводит к компилированию файла *sum_before_and_after_negative.c*. Для исполнения цели необходимы файлы *sum_before_and_after_negative.c, sum_before_and_after_negative.h, index_first_negative.h, index_last_negative.h*.

**Выводы.**

В ходе выполнения лабораторной работы было изучено, как производить сборку программ в языке программирования Си. Каждая функция программы была вынесена в отдельный файл исходного кода. Для каждого файла исходного кода функции был написан заголовочный файл. Был написан Makefile для компиляции программы.

Разработана программа, считывающая с клавиатуры входные данные и команды пользователя, и в зависимости от выбранной команды вызывающая одну из четырех функций:

0 : функция возвращает индекс первого отрицательного элемента.

1 : функция возвращает индекс последнего отрицательного элемента.

2 : функция возвращает сумму модулей элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент).

3 : функция возвращает сумму модулей элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент).

**Тестирование.**

Данные тестирования представлены в таблице 1.

Таблица 1 – Результаты тестирования

| № 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 | Результат корректен |
| 2. | 1 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11 | 20 | Результат корректен |
| 3. | 2 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11 | 226 | Результат корректен |
| 4. | 3 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11 | 30 | Результат корректен |
| 5. | 0 17 9 7 20 21 -28 20 12 -24 15 -6 5 11 4 27 | 5 | Результат корректен |
| 6. | 1 -22 23 22 -28 4 6 18 21 25 19 19 -28 9 26 0 -15 11 25 12 | 15 | Результат корректен |
| 7. | 2 26 22 27 26 11 -18 25 4 2 1 22 -9 19 26 19 15 -27 -4 0 29 6 21 2 | 187 | Результат корректен |
| 8. | 3 22 -19 11 24 7 24 -21 -17 9 29 11 15 25 22 17 7 26 24 10 2 14 | 250 | Результат корректен |

## ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: index_first_negative.h

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

Название файла: index_last_negative.h

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

Название файла: sum_between_negative.h

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

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

Название файла: sum_before_and_after_negative.h

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

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

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 DEFAULT_SIZE 100

int main() {
	int function_index;
	char last_character;
	scanf("%d%c", &function_index, &last_character);

	int numbers[DEFAULT_SIZE];
	int array_size = 0;

	for (int i = 0; last_character != '\n'; i++) {
		scanf("%d%c", &numbers[i], &last_character);
		array_size++;
	}

	switch (function_index) {
		case 0:
			printf("%d\n", index_first_negative(numbers,
array_size));
			break;
		case 1:
```

```
                                printf("%d\n", index_last_negative(numbers,
array_size));
                                break;
                        case 2:
                                printf("%d\n", sum_between_negative(numbers,
array_size));
                                break;
                        case 3:
                                printf("%d\n",
sum_before_and_after_negative(numbers, array_size));
                                break;
                        default:
                            printf("Данные некорректны");
                }
                return 0;
}
```

Название файла:  index_first_negative.c

```
#include "index_first_negative.h"

int index_first_negative(int numbers[], int array_size) {
          for (int i = 0; i < array_size; i++) {
                    if (numbers[i] < 0){
                              return i;
                    }
          }
}
```

Название файла: index_last_negative.c

```
#include "index_last_negative.h"

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

Название файла: sum_between_negative.c

```
#include "sum_between_negative.h"

int sum_between_negative(int numbers[], int array_size) {
          int sum = 0;
          for (int i = index_first_negative(numbers, array_size); i <
index_last_negative(numbers, array_size); i++) {
                    sum += abs(numbers[i]);
          }
          return sum;
}
```

Название файла:  sum_before_and_after_negative.c

```
#include "sum_before_and_after_negative.h"

int sum_before_and_after_negative(int numbers[], int array_size) {
	int sum = 0;
	for (int i = 0; i < index_first_negative(numbers, array_size); i++) {
		sum += abs(numbers[i]);
	}
	for (int i = index_last_negative(numbers, array_size); i < array_size;
i++) {
		sum += abs(numbers[i]);
	}
	return sum;
}
```

Название файла: Makefile

```
all: menu.o index_first_negative.o index_last_negative.o sum_between_negative.o
sum_before_and_after_negative.o index_first_negative.h index_last_negative.h
sum_between_negative.h sum_before_and_after_negative.h
	gcc menu.o index_first_negative.o index_last_negative.o
sum_between_negative.o sum_before_and_after_negative.o -o menu

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

index_first_negative.o: index_first_negative.c index_first_negative.h
	gcc -c index_first_negative.c

index_last_negative.o: index_last_negative.c index_last_negative.h
	gcc -c index_last_negative.c

sum_between_negative.o: sum_between_negative.c sum_between_negative.h
index_first_negative.h index_last_negative.h
	gcc -c sum_between_negative.c

sum_before_and_after_negative.o: sum_before_and_after_negative.c
sum_before_and_after_negative.h index_first_negative.h index_last_negative.h
	gcc -c sum_before_and_after_negative.c

clean:
	rm menu *.o
```