

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 проектов в языке СИ

Студент гр. 2384

Шурыгин Д. Л.

Преподаватель

Гаврилов А. В.

Санкт-Петербург

Цель работы.

Изучить процесс сборки программ, написанных на языке C, с помощью make-файлов.

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 20**. 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 произведение элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент). (multi_between_negative.c)
- 3 : Найти произведение элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент).
(multi_before_and_after_negative.c)

иначе необходимо вывести строку "Данные некорректны".

Ошибкой в данном задании считается дублирование кода!

Подсказка: функция нахождения модуля числа находится в заголовочном файле `stdlib.h` стандартной библиотеки языка Си. При выводе результата, не забудьте символ переноса строки

Выполнение работы.

Создаются .c файлы, в которых определяются функции и заголовочные файлы с объявлением каждой из функций. В заголовочных файлах определяются функции и необходимые зависимости для некоторых функций, включается препроцессорная директива `#pragma once`, чтобы конкретный файл при компиляции подключался строго один раз.

В файле `menu` происходит считывание данных через функцию `scanf`, которые вводит пользователь, число, которое отвечает за выполняемую функцию программы и массив целых чисел, количество которых не превышает 20. Все числа разделены пробелами.

По итогу вычисляется `n` – количество чисел в массиве, записывается сам массив `array`, и переменная `dev`, которая отвечает за функции программы в теле `main`. Также подключаются заголовочные файлы `index_first_negative.h`, `index_last_negative.h`, `multi_between_negative.h`, `multi_before_and_after_negative.h` и `stdio.h`.

Функции `index_first_negative` на вход подаётся массив `array` и количество чисел в массиве `n`, возвращает функция индекс первого отрицательного числа в массиве, которое находится с помощью цикла `for`.

Функции `index_last_negative` на вход подаётся массив `array` и количество чисел в массиве `n`, возвращает функция индекс последнего

отрицательного числа в массиве, которое также находится с помощью цикла for.

Функции `multi_between_negative` на вход подаётся массив `array` и количество чисел в массиве `n`, функция возвращает произведение элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент) с помощью цикла for.

Функции `multi_before_and_after_negative` на вход подаётся массив `array` и количество чисел в массиве `n`, функция возвращает произведение элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент) с помощью двух циклов for.

Далее в теле функции `menu` реализуется оператор `switch` с помощью переменной `dev`, значения которой подаются в `case`:

- Если `case = 0`, то вызывается функция `index_first_negative`, которая возвращает индекс первого отрицательного элемента в массиве;
 - Если `case = 1`, то вызывается функция `index_last_negative`, которая возвращает индекс последнего отрицательного элемента в массиве;
 - Если `case = 2`, то вызывается функция `multi_between_negative`, и возвращает соответствующее ей произведение;
 - Если `case = 3`, то вызывается функция `multi_before_and_after_negative`, и возвращает соответствующее ей произведение.
- Если `case` не принадлежит от 0 до 3 включительно, то программа выведет сообщение “Данные некорректны”.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

	№Входные данные	Выходные данные	Комментарии
1.	0 -5 -3 -5 -8 3 -9 -3	0	Первый отрицательный элемент -5, его индекс равен 0.
2.	1 -5 -3 -5 -8 3 -9 -3	6	Последний отрицательный элемент -3, его индекс равен 6.
3.	2 -5 -3 -5 -8 3 -9 -3	-16200	Произведение чисел от -5 включая до -3 не включая
4.	3 -5 -3 -5 -8 3 -9 -3	-3	Так как индекс -5 равен 0, то произведение только -3

Выводы.

Была изучена утилита Make, создание make-файлов и конечная сборка проектов. Разработан Makefile собирающий и компилирующий программу, которая состоит из нескольких файлов. Компилируется

программа с помощью утилиты Make, пишется make-файл со всеми зависимостями.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: menu.c

```
#include <stdio.h>
#include "index_first_negative.h"
#include "index_last_negative.h"
#include "multi_between_negative.h"
#include "multi_before_and_after_negative.h"
#define N 20

int main(){
    int array[N];
    int n = 0;
    int i;
    char c;
    int dev;
    scanf("%d", &dev);
    do{
        scanf("%d%c", &array[n], &c);
        n = n + 1;
    }while(n < N && c != '\n');
    switch (dev)
    {
    case 0:
        printf("%d\n",index_first_negative(n, array));
        break;

    case 1:
        printf("%d\n",index_last_negative(n, array));
        break;

    case 2:
        printf("%d\n",multi_between_negative(n, array));
        break;

    case 3:
        printf("%d\n",multi_before_and_after_negative(n, array));
        break;

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


Название файла: index_first_negative.c

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

```
int index_first_negative(int n, int array[]){
    int first_neg;
    int i;
    for (i = 0; i < n; i++){
        if (array[i] < 0){
            first_neg = i;
            return first_neg;
            break;
        }
    }
}
```

Название файла: index_first_negative.h

```
#pragma once
```

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

Название файла: index_last_negative.c

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

```
int index_last_negative(int n, int array[]){
    int last_neg;
    int i;
    for (i = 0; i < n; i++){
        if (array[i] < 0){
            last_neg = i;
        }
    }
    return last_neg;
}
```

Название файла: index_last_negative.h

```
#pragma once
```

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

Название файла: multi_before_and_after_negative.c

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

int multi_before_and_after_negative(int n, int array[]){
    int pro_res,i,l,k;
    pro_res = 1;
    i = index_first_negative(n, array);
    l = index_last_negative(n, array);
    for (k = 0; k < i; k++){
        pro_res = pro_res * array[k];
    }
    for (l ; l < n; l++){
        pro_res = pro_res * array[l];
    }
    return pro_res;
}
```

Название файла: multi_before_and_after_negative.h

```
#pragma once
int multi_before_and_after_negative(int n, int array[]);
```

Название файла: multi_between_negative.c

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

int multi_between_negative(int n, int array[]){
    int pro_res,i,l;
    pro_res = 1;
    i = index_first_negative(n, array);
    l = index_last_negative(n, array);
    for (i; i < l; i++){
        pro_res = pro_res * array[i];
    }
    return pro_res;
}
```

Название файла: multi_between_negative.h

```
#pragma once
int multi_between_negative(int n, int array[]);
```

Название файла: Makefile

```
all: menu.o index_first_negative.o index_last_negative.o
multi_before_and_after_negative.o multi_between_negative.o
    gcc menu.o index_first_negative.o index_last_negative.o
multi_between_negative.o multi_before_and_after_negative.o -o menu
main.o: menu.c index_first_negative.h index_last_negative.h
multi_before_and_after_negative.h multi_between_negative.h
    gcc -c -std=c99 menu.c
index_first_negative.o: index_first_negative.c index_first_negative.h
    gcc -c -std=c99 index_first_negative.c
index_last_negative.o: index_last_negative.c index_last_negative.h
    gcc -c -std=c99 index_last_negative.c
multi_before_and_after_negative.o: multi_before_and_after_negative.c
multi_before_and_after_negative.h
    gcc -c -std=c99 multi_before_and_after_negative.c
multi_between_negative.o: multi_between_negative.c
multi_between_negative.h
    gcc -c -std=c99 multi_between_negative.c
```