

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. 1304

Ха Ди Ен

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Научиться создавать проект с make-файлом для программы, состоящей из нескольких файлов, и собирать его.

Задание.

Вариант № 1.

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".

Выполнение работы.

Переменные, созданные и используемые в ходе выполнения программы:

1. В функции `main`:

- `arr[]` – целочисленный массив размера не больше `N` (объявлен через `#define N 20`)
- `a` – целочисленное значение, принимаемое на вход и определяющее последующую исполняемую функцию
- `k` – целочисленная переменная, которая считывает количество введённых в массив `arr[]` чисел (значение по умолчанию – 0)
- `i` – целочисленная переменная, которая считывает индексы элементов массива `arr[]` (шаг по массиву)
- `s` – символьная переменная, считывающая символ после каждого введённого числа в массиве `arr[]`

2. В остальных функциях:

- `a[]` – целочисленный массив размера не больше `N` (объявлен через `#define N 20`)
- `i` – целочисленная переменная, которая считывает индексы элементов массива `a[]` (шаг по массиву)
- `n` – целочисленная переменная, которая считывает количество введённых в массив `a[]` чисел (значение по умолчанию – 0)
- `fn` – целочисленная переменная, присваивающая значение индекса первого отрицательного элемента (функция `index_first_negative`)
- `ln` – целочисленная переменная, присваивающая значение индекса последнего отрицательного элемента (функция `index_last_negative`)
- `mb` – целочисленная переменная, присваивающая значение произведения элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего

отрицательного (не включая элемент) (функция *multi_between_negative*)

- *mba* – целочисленная переменная, присваивающая значение произведения элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент) (функция *multi_before_and_after_negative*)

Функции, реализованные в программе:

- Функция *index_first_negative(int a[], int n)* принимает на вход массив *a[]* и переменную *n* (длина массива), после чего с помощью цикла *for* получает индекс первого отрицательного элемента и возвращает его.
- Функция *index_last_negative(int a[], int n)* принимает на вход массив *a[]* и переменную *n* (длина массива), после чего с помощью цикла *for* получает индекс последнего отрицательного элемента и возвращает его.
- Функция *multi_between_negative(int a[], int n)* принимает на вход массив *a[]* и переменную *n* (длина массива), после чего с помощью цикла *for* находит произведение элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент) и возвращает его.
- Функция *multi_before_and_after_negative(int a[], int n)* принимает на вход массив *a[]* и переменную *n* (длина массива), после чего с помощью цикла *for* находит произведение элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент) и возвращает его.

Каждая из 4 функций хранится в отдельном файле с соответствующим названием (расширение .c) и имеет прототип, вынесенный в отдельный файл (расширение .h). С помощью заголовочных файлов осуществляется

линковка данных функций в одну программу (исполняемый файл – menu) с помощью make-файла (Makefile).

- Функция *main()* принимает на вход значение *a* и массив *arr[]* длиной *k* ($k \leq N$) (значения массива считываются с помощью цикла *for*). Затем с помощью оператора *switch* соответственно каждому значению *a* – 0-3 (если *a* принимает другие значения, то выводится строка «Данные некорректны») выполняется одна из подзадач путём вызова функции и выводится полученное в функции значение.

Файл, который реализует главную функцию (*main*), называется menu.c.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 -2 3 4 -2 5 -3 2 4	0	Вызывается функция <code>index_first_negative</code> , которая возвращает индекс первого отрицательного элемента.
2.	1 -2 3 4 -2 3 -2	5	Вызывается функция <code>index_last_negative</code> , которая возвращает индекс последнего отрицательного элемента.
3.	2 -3 2 1 -2 3 -2 1 2 3	36	Вызывается функция <code>multi_between_negative</code> , которая возвращает произведение элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент).
4.	3 4 3 -55 17 43 -31 4 -2 1 4 2	-192	Вызывается функция <code>multi_before_and_after_negative</code> , которая возвращает произведение элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент).
5.	4 3 2 5 3 -4	Данные некорректны	Поданное значение не совпало ни с одним из предложенных (0-3).

Выводы.

Мы научились создавать проект с make-файлом для программы, состоящей из нескольких файлов, и собирать его.

Написан make-файл программы, выполняющей считывание с клавиатуры исходных данных и команды пользователя. Для линковки и компиляции программы, состоящей из нескольких файлов, созданы заголовочные файлы с прототипами функций.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: menu.c

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

#define N 20

int main()
{
    char s;
    int arr[N], a, k = 0, i;
    scanf("%d", &a);
    for (i = 0; i < N; i++) {
        scanf("%d%c", &arr[i], &s);
        k++;
        if (s == '\n')
            break;
    }

    switch (a) {
        case 0:
            printf("%d\n", index_first_negative(arr, k));
            break;
        case 1:
            printf("%d\n", index_last_negative(arr, k));
            break;
        case 2:
            printf("%d\n", multi_between_negative(arr, k));
            break;
        case 3:
            printf("%d\n", multi_before_and_after_negative(arr, k));
            break;
        default:
            puts("Данные некорректны\n");
    }
    return 0;
}
```

Название файла: index_first_negative.h

```
#include <stdio.h>

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

Название файла: index_first_negative.c

```

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

int index_first_negative(int a[], int n)
{
    int fn = 0, i;
    for (i = n - 1; i >= 0; i--) {
        if (a[i] < 0) {
            fn = i;
        }
    }

    return fn;
}

```

Название файла: index_last_negative.h

```

#include <stdio.h>

int index_last_negative(int a[], int n);

```

Название файла: index_last_negative.c

```

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

int index_last_negative(int a[], int n)
{
    int ln = n - 1, i;
    for (i = 0; i < n; i++) {
        if (a[i] < 0) {
            ln = i;
        }
    }

    return ln;
}

```

Название файла: multi_before_and_after_negative.h

```

#include <stdio.h>

int multi_before_and_after_negative(int a[], int n);

```

Название файла: multi_before_and_after_negative.c

```

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

int multi_before_and_after_negative(int a[], int n)
{
    int mba = 1, i;
    for (i = 0; i < index_first_negative(a, n); i++) {
        mba *= a[i];
    }
    for (i = index_last_negative(a, n); i < n; i++) {
        mba *= a[i];
    }

    return mba;
}

```

Название файла: multi_between_negative.h

```

#include <stdio.h>

int multi_between_negative(int a[], int n);

```

Название файла: multi_between_negative.c

```

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

int multi_between_negative(int a[], int n)
{
    int mb = 1, i;
    for (i = index_first_negative(a, n); i < index_last_negative(a,
n); i++) {
        mb *= a[i];
    }

    return mb;
}

```

Название файла: Makefile

```

all:          menu.o          multi_before_and_after_negative.o
multi_between_negative.o index_last_negative.o index_first_negative.o

```

```

        gcc            menu.o            multi_before_and_after_negative.o
multi_between_negative.o index_last_negative.o index_first_negative.o -o
menu
        menu.o:            menu.c            multi_before_and_after_negative.h
multi_between_negative.h index_last_negative.h index_first_negative.h
        gcc -c menu.c
        mba.o:            multi_before_and_after_negative.c
multi_before_and_after_negative.h            index_last_negative.h
index_first_negative.h
        gcc -c multi_before_and_after_negative.c
        mb.o:            multi_between_negative.c            multi_between_negative.h
index_last_negative.h index_first_negative.h
        gcc -c multi_between_negative.c
        ln.o: index_last_negative.c index_last_negative.h
        gcc -c index_last_negative.c
        fn.o: index_first_negative.c index_first_negative.h
        gcc -c index_first_negative.h
clean:
        rm -rf *.o menu

```