

**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**

Студент гр. 1304

Ефремов А.А.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

## **Цель работы.**

Научиться создавать проект с make-файлом и собирать проект

## **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.**

Проект состоит из файлов:

`menu.c` — реализует главную функцию `main`, которая в зависимости от входа выполняет необходимую операцию над числами и выводит результат в консоль.

`index_first_negative.c` — реализует функцию `index_first_negative`, которая принимает на вход массив чисел и его размер и ищет индекс первого отрицательного элемента в массиве.

`index_first_negative.h` — объявление функции `index_first_negative`.

`index_last_negative.c` — реализует функцию `index_last_negative`, которая принимает на вход массив чисел и его размер и ищет индекс последнего отрицательного элемента в массиве.

`index_last_negative.h` — объявление функции `index_last_negative`.

`multi_between_negative.c` — реализует функцию `multi_between_negative`, которая принимает на вход массив чисел, его размер и указатель на переменную `err`, отвечающую за проверку корректности введенных данных, и возвращает произведение элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент).

`multi_between_negative.h` — объявление функции `multi_between_negative`.

`multi_before_and_after_negative.c` — реализует функцию `multi_before_and_after_negative`, которая принимает на вход массив чисел, его размер и указатель на переменную `err`, отвечающую за проверку корректности введенных данных, и возвращает произведение элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент)

`multi_before_and_after_negative.h` — объявление функции `multi_before_and_after_negative`.

`Makefile` — файл сборки программы.

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 -1 2 3 4 5 6 7 9 1 2	0	Верно
2.	1 -1 -2 -3 9 8 9 3 0 -4	8	Верно
3.	2 9 8 -1 6 7 -3 4 5 6 7	-42	Верно
4.	3 4 5 -1 3 4 5 -6 -6 -5 7	-700	Верно
5.	4 9 8 0 3 1 7 3 5 8 -4	Данные некорректны	Верно

### **Выводы.**

Написан Makefile и собран проект с программой, реализующей ввод и вывод данных. Все функции проекта были определены в отдельных файлах. В процессе сборки проекта были написаны заголовочные файлы для определенных функций.

## 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"
int main()
{
    int i;
    int fv, err = 0, ans = 0, counter = 0;
    int arr[20] = { 0 };
    char ch;
    scanf("%d", &fv);
    for (i = 0; i < 20; i++)
    {
        scanf("%d%c", &arr[i], &ch);
        counter++;
        if (ch == '\n')
            break;
    }
    switch (fv)
    {
    case 0:
        ans = index_first_negative(arr, counter);
        if (ans != -1)
            printf("%d\n", ans);
        else
            printf("Данные некорректны\n");
        break;
    case 1:
        ans = index_last_negative(arr, counter);
        if (ans != -1)
            printf("%d\n", ans);
        else
            printf("Данные некорректны\n");
        break;
    case 2:
        ans = multi_between_negative(arr, counter, &err);
        if (err)
            printf("Данные некорректны\n");
        else
            printf("%d\n", ans);
        break;
    case 3:
        ans = multi_before_and_after_negative(arr, counter,
&err);
        if (err)
            printf("Данные некорректны\n");
        else
```

```

        printf("%d\n", ans);
        break;
default:
    printf("Данные некорректны\n");
    break;
}
return 0;
}

```

Название файла: index\_first\_negative.h

```

#pragma once
int index_first_negative(int arr[], int N);

```

Название файла: index\_first\_negative.c

```

#include <stdio.h>
#include "index_first_negative.h"
int index_first_negative(int arr[], int N)
{
    int i;
    for (i = 0; i < N; i++)
        if (arr[i] < 0)
            return i;
    return -1;
}

```

Название файла: index\_last\_negative.h

```

#pragma once
int index_last_negative(int arr[], int N);

```

Название файла: index\_last\_negative.c

```

#include <stdio.h>
#include "index_last_negative.h"
int index_last_negative(int arr[], int N)
{

```

```

    int i;
    int index_l = -1;
    for (i = 0; i < N; i++)
        if (arr[i] < 0)
            index_l = i;
    return index_l;
}

```

Название файла: multi\_between\_negative.h

```

#pragma once
int multi_between_negative(int arr[], int N, int *err);

```

Название файла: multi\_between\_negative.c

```

#include <stdio.h>
#include "index_first_negative.h"
#include "index_last_negative.h"
#include "multi_between_negative.h"
int multi_between_negative(int arr[], int N, int *err)
{
    int i;
    int res = 1;
    int index_f = index_first_negative(arr, N);
    int index_l = index_last_negative(arr, N);
    if ((index_f == index_l) || (index_f < 0) || (index_l < 0))
    {
        *err = 1;
        return 0;
    }
    for (i = index_f; i < index_l; i++)
        res *= arr[i];
    return res;
}

```

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

```

#pragma once
int multi_before_and_after_negative(int arr[], int N, int *err);
Название файла: multi_before_and_after_negative.c
#include <stdio.h>
#include "index_first_negative.h"
#include "index_last_negative.h"
#include "multi_before_and_after_negative.h"
int multi_before_and_after_negative(int arr[], int N, int *err)
{
    int i;
    int res = 1;
    int index_f = index_first_negative(arr, N);
    int index_l = index_last_negative(arr, N);
    if ((index_f == index_l) || (index_f < 0) || (index_l < 0))
    {
        *err = 1;
        return 0;
    }
    for (i = 0; i < index_f; i++)
        res *= arr[i];
    for (i = index_l; i < N; i++)
        res *= arr[i];
    return res;
}

```

## Название файла: Makefile

```

all: menu

menu:    menu.o    index_first_negative.o    index_last_negative.o
multi_between_negative.o multi_before_and_after_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
menu.o:  menu.c    index_first_negative.h    index_last_negative.h
multi_between_negative.h multi_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
        multi_between_negative.o:                            multi_between_negative.c
multi_between_negative.h index_first_negative.h index_last_negative.h
        gcc -c multi_between_negative.c
        multi_before_and_after_negative.o:
multi_before_and_after_negative.c      multi_before_and_after_negative.h
index_first_negative.h index_last_negative.h
        gcc -c multi_before_and_after_negative.c
clean:
        rm -rf *.o menu

```