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 _____ Лобанов Е.А.

Преподаватель _____ Чайка К.В.

Санкт-Петербург
2021

Цель работы.

Научиться создавать программу, состоящую из нескольких файлов. Научиться создавать заголовочные файлы и make-файлы и оперировать ими.

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".

Основные теоретические положения.

Makefile – список инструкций для утилиты *make*, которая позволяет собирать проект сразу целиком. Состоит из списка целей, зависимостей и команд:

```
цель: зависимости
[tab] команда
```

#include – вызывает заголовочный файл в текущий файл исходного кода
#define – позволяет определить макросы или макроопределения.

Выполнение работы.

В ходе выполнения задания были использованы переменные:

Целочисленная переменная *task* считывает в себя номер задания.

Массив *arr[N]* длины *N* (объявлен через *#define N 20*) содержит в себе целые числа.

Целочисленная переменная *count* считывает количество введённых значений в массив *arr[N]*, значение по умолчанию – 0.

Символьная переменная *c* считывает символ, введённый после каждого введённого числа.

Целочисленная переменная *indx* считывает индексы чисел для для функций *index_first_negative* и *index_last_negative.*

Целочисленная переменная *result* присваивает себе значение вычислений функций *multi_between_negative* и *multi_before_and_after_negative.*

Целочисленная переменная *first_negative* и *last_negative* принимают себе значения функций *index_first_negative* и *index_last_negative* соответственно.

Для выполнения каждой из подзадач были созданы 4 различных прототипа функций, вынесенных в отдельные файлы и вызываемых с помощью заголовочных файлов, и основная функция main:

Функция *main* в файле *main.c* производит номер задания *task,* ввод массива *arr[N]* и подсчёт количества введённых элементов *count* данного массива, затем с помощью оператора *switch* соответственно каждому значению *task* от 0 до 3 выполняет одну из подзадач с помощью вызова функций, если значение *task* не равно вышеописанным значениям, то выводится строка "Данные некорректны"

Функция *index_first_negative* получает на вход массив *arr[]* и количество элементов в массиве count. Затем, с помощью цикла for перебирает элементы массива, и возвращает индекс первого отрицательного элемента.

Функция *index_last_negative* получает на вход массив *arr[]* и количество элементов в массиве count. Затем, с помощью цикла for перебирает элементы массива, и возвращает индекс последнего отрицательного элемента.

Функция *multi_between_negative* получает на вход массив *arr[]* и количество элементов в массиве count. Затем функция получает значения функций *index_first_negative* и *index_last_negative* и затем осуществляет вывод результата задачи, подсчитанного с помощью цикла for результата перемножения всех элементов от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент).

Функция *multi_before_and_after_negative* получает на вход массив *arr[]* и количество элементов в массиве count. Затем функция получает значения функций *index_first_negative* и *index_last_negative* и затем осуществляет вывод результата задачи, подсчитанного с помощью цикла for результата перемножения всех элементов до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент).

Каждая из четырёх вышеперечисленных функций хранится в отдельном в файле с соответственным названием. С помощью заголовочных файлов для каждой из этих функций осуществляется линковка этих функций в одну программу под названием *menu* с помощью make-файла

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 3 90 -7 239 -444 123	3	Верно
2.	1 1 3 90 -7 239 -444 123	5	Верно

3.	2 1 3 90 -7 239 -444 123	-1673	Верно
4.	3 1 3 90 -7 239 -444 123	-14745240	Верно
5.	4 1 3 90 -7 239 -444 123	Данные некорректны	Верно

Выводы.

Мы научились компилировать и компоновать программу, состоящую из нескольких файлов. Освоили работу с заголовочными файлами и make-файлами.

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 arr[N], task, count = 0;
    char c;
    scanf("%d%c", &task, &c);
    for (int i = 0; c != '\n'; i++){
        scanf("%d%c", &arr[i], &c);
        count++;
    }
    switch (task) {
        case 0:
            printf("%d\n", index_first_negative(arr, count));
            break;
        case 1:
            printf("%d\n", index_last_negative(arr, count));
            break;
        case 2:
            printf("%d\n", multi_between_negative(arr, count));
            break;
        case 3:
            printf("%d\n", multi_before_and_after_negative(arr, count));
            break;
        default:
            puts("Данные некорректны");
    }
    return 0;
}
```

Название файла: index_first_negative.c

```
#include "index_first_negative.h"

int index_first_negative(int arr[], int count){
    int indx;
    for (int i = 0; i < count; ++i){
        if (arr[i] < 0){
            indx = i;
            break;
        }
    }
    return indx;
}
```

Название файла: index_first_negative.h

```
int index_first_negative(int arr[], int count);
```

Название файла: index_last_negative.c

```
#include "index_last_negative.h"

int index_last_negative(int arr[], int count){
    int indx;
    for (int i = count - 1; i >= 0; --i){
        if (arr[i] < 0){
            indx = i;
            break;
        }
    }
    return indx;
}
```

Название файла: index_last_negative.h

```
int index_last_negative(int arr[], int count);
```

Название файла: multi_between_negative.c

```
#include "multi_between_negative.h"
#include "index_first_negative.h"
#include "index_last_negative.h"

int multi_between_negative(int arr[], int count){
    int result = 1, first_negative = index_first_negative(arr, count),
last_negative = index_last_negative(arr, count);
    for(int i = first_negative; i < last_negative; ++i){
        result *= arr[i];
    }
    return result;
}
```

Название файла: multi_between_negative.h

```
int multi_between_negative(int arr[], int count);
```

Название файла: multi_before_and_after_negative.c

```
#include "multi_before_and_after_negative.h"
#include "index_first_negative.h"
#include "index_last_negative.h"

int multi_before_and_after_negative(int arr[], int count){
    int result = 1, first_negative = index_first_negative(arr,
count), last_negative = index_last_negative(arr, count);
    for(int i = 0; i < first_negative; ++i){
        result *= arr[i];
    }
    for(int i = last_negative; i < count; ++i){
        result *= arr[i];
    }
    return result;
}
```

Название файла: multi_before_and_after_negative.h

```
int multi_before_and_after_negative(int arr[], int count);
```

Название файла: Makefile

```
all: 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
index_first_negative.h index_last_negative.h multi_between_negative.h
	gcc -c multi_between_negative.c

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

clean:
	rm -rf *.o menu
```