

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

Арчибасов Е.О.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Научиться использовать условия, циклы и оператор switch в языке программирования C.

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**, с помощью которой можно собирать цельный проект. Имеет вид:

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

Выполнение работы.

Для выполнения каждой из подзадач были созданы 4 различных функции и основная функция `main`:

Функция **`main`** считывает с экрана вводимое значение и записывает его в переменную `a`. Затем она с помощью цикла **`for`** и вложенного в него цикла **`if`** считывает целое число и идущий за ним символ, пока не дойдет до символа переноса строки, таким образом получает количество элементов в введенном массиве, как номер индекса, на котором цикл прервется, плюс единица, так как индекс первого элемента равен нулю. Целые числа записываются в массив под соответствующим индексом. Затем с помощью оператора **`switch`** каждому значению переменной `a` от 0 до 3 соответствует определенная функция, выполняющая определенную подзадачу. Если значение `a` принимает другие значения, то на экран выводится сообщение "Данные некорректны".

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

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

Функция **`multi_between_negative`** получает на вход массив `Arr[]` и

количество элементов в массиве *len*. Затем осуществляет возвращение результата задачи, подсчитанного с помощью цикла *for*, в котором исходное значение и значение выхода из цикла получаем с помощью функций *index_first_negative* и *index_last_negative*, и который получен перемножением 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 *len*. Затем осуществляет возвращение результата задачи, подсчитанного с помощью цикла *for*, в котором исходное значение и значение выхода из цикла получаем с помощью функций *index_first_negative* и *index_last_negative*, и который получен перемножением всех элементов до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент).

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 -5 7 8 2 1 12 -9 2	1	Верно
2.	1 15 49 -8 92 15 48 -9 21 48	6	Верно
3.	2 54 -7 1 2 1 1 -8 1 4 87 14	-14	Верно
4.	3 1 2 3 -4 5 1 4 7 6 -10 2 1	-120	Верно
5.	8 1 4 5 2 4 4 -4 27 -9 12 3	Данные некорректны	Верно

Вывод:

В ходе выполнения работы мы на практике столкнулись с такими понятиями, как компиляция, компоновка/линковка. Научились разбивать программу на части и работать с заголовочными и make файлами.

