МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №1
по дисциплине «Программирование»
Тема: Условия, циклы, оператор switch

Студент гр. 1304 _____ Лобанов Е.А.

Преподаватель _____ Чайка К.В.

Санкт-Петербург
2021

Цель работы.

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

Задание.

Вариант 1.

Напишите программу, выделив каждую подзадачу в отдельную функцию.

Реализуйте программу, 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*)

1 : индекс последнего отрицательного элемента. (*index_last_negative*)

2 : Найти произведение элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент). (*multi_between_negative*)

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

Иначе необходимо вывести строку "Данные некорректны".

Основные теоретические положения.

Условный оператор if-else:

```
if (<условие>){
      <блок действий при истинности условия>;
} else {
      <Блок действий при ложности условия>;
}
```

Условный оператор switch:

```
switch (<условие>) {
      case (<значение 1>):
            <блок действий 1>;
      case (<значение 2>):
            <блок действий 2>;
```

```
…
case (<значение n>):
      <блок действий n>;
default:
      <блок действий если ни одно значение не равно условию>;
```

Цикл for:

```
for(<исходное значение условия>; <значение выхода из цикла>; <изменение условия>){
      <блок действий>
}
```

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 функции, а также функция ввода массива и основная функция main:

Функция *read_arr* производит ввод массива *arr[N]* и подсчёт количества введённых элементов *count* данного массива.

Функция *main* получает на вход от функции *read_arr* массив *arr* и количество элементов *count* этого 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).

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.

Были изучены условия, циклы и оператор switch в языке программирования C.

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя и обрабатывающая массивы. Для обработки команд пользователя использовались условные операторы *if-else*, *switch*, а также цикл for.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>

#define N 20

int read_arr(int arr[], int length){
    char c;
    int count;
    for (int i = 0; i < length; ++i){
        scanf("%d%c", &arr[i], &c);
        count++;
        if (c == '\n'){
            break;
        }
    }
    return count;
}

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;
}

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;
}

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;
}

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;
    }

    int main(){
        int arr[N], task, count = 0;
        char c;
        scanf("%d%c", &task, &c);
        count = read_arr(arr, N);
        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;
}
```