

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

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.

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.

stdio.h – Стандартная библиотека ввода и вывода.

printf(<строка форматов>, арг 1, арг 2, ..., арг n) - Стандартная функция, используемая для форматируемого вывода

scanf(<строка форматов>, арг 1, арг 2, ..., арг n) - Стандартная функция, используемая для форматируемого ввода

Условный оператор **if-else**:

IF (<УСЛОВИЕ>) {

```
    <ТЕЛО>;  
} ELSE {  
    <ТЕЛО >;  
}
```

Условный оператор **switch**:

```
SWITCH (<УСЛОВИЕ>) {  
    CASE (<ЗНАЧЕНИЕ 1>):  
        <ТЕЛО >;  
    CASE (<ЗНАЧЕНИЕ 2>):  
        <ТЕЛО >;  
    ...  
    CASE (<ЗНАЧЕНИЕ N>):  
        <ТЕЛО >;  
    DEFAULT:  
        <ТЕЛО >;  
}
```

Цикл **while**:

```
WHILE (УСЛОВИЕ) {  
    <ТЕЛО >;  
}
```

Цикл **for**:

```
FOR (<ИСХОДНОЕ ЗНАЧЕНИЕ УСЛОВИЯ>; <ЗНАЧЕНИЕ ВЫХОДА ИЗ ЦИКЛА>;  
<ИЗМЕНЕНИЕ УСЛОВИЯ>) {  
    <ТЕЛО >  
}
```

Выполнение работы.

В ходе выполнения задания были использованы переменные:

- Целочисленная переменная a принимает значение номера задания
- Массив Arr[N] с максимальной длиной N содержит целые числа

(объявлена через *#define N 20*)

- Целочисленная переменная *len* предназначена для того, чтобы считать количество введенных в массив *Arr[]* значений
- Символьная переменная *n* считывает символ, следующий за каждым введенным числом
- Целочисленная переменная *i* используется независимо в двух функциях (*index_first_negative* и *index_last_negative*). В обеих функциях она считывает индекс элемента массива *Arr[]* и является возвращаемым значением
- Целочисленная переменная *multi_between* присваивает себе значение вычисления функции *multi_between_negative*
- Целочисленная переменная *multi_before_and_after* присваивает себе значение вычисления функции *multi_before_and_after_negative*

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 1 -5 4 -9 2	2	Верно
2.	1 15 49 -8 92 -9 21 48	4	Верно
3.	2 54 -7 1 1 -8 1	-7	Верно

4.	3 14 14 -5 1 -7 1	-1372	Верно
5.	4 14 14 141 -4 -8 14	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* и *while*.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: 1304_Archibasov_EO_PR.c

```
#include <stdio.h>

#define N 20

int index_first_negative(int array[], int length){
    int i = 0;
    while(array[i] >= 0){
        i++;
    }
    return i;
}

int index_last_negative(int array[], int length){
    int i = length - 1;
    while(array[i] >= 0){
        i--;
    }
    return i;
}

int multi_between_negative(int array[], int length){
    int multi_between = 1;
    for(int i = index_first_negative(array, length); i <
index_last_negative(array, length); i++){
        multi_between *=array[i];
    }
    return multi_between;
}

int multi_before_and_after_negative(int array[], int length){
    int multi_before_and_after = 1;
```

```

    for(int i = 0; i < index_first_negative(array, length); i++){
        multi_before_and_after*=array[i];
    };
    for(int i = index_last_negative(array, length); i < length ; i++){
        multi_before_and_after*=array[i];
    };
    return multi_before_and_after;
}

int main(){
    int a, len, Arr[N];
    char n;

    scanf("%d\n", &a);

    for(int i = 0; i < N; i++){

        scanf("%d%c", &Arr[i], &n);
        if(n == '\n'){
            len = i + 1;
            break;
        }
    }

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

```