

**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**

Студентка гр. 1304

Ха Ди Ен

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

### **Цель работы.**

Изучить управляющие конструкции языка С такие, как условия, циклы и оператор switch, и научиться применять их для написания простейших программ.

### **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".

## Выполнение работы.

Переменные, созданные и используемые в ходе выполнения программы:

1. В функции `main`:

- `arr[]` – целочисленный массив размера не больше `N` (объявлен через `#define N 20`)
- `a` – целочисленное значение, принимаемое на вход и определяющее последующую исполняемую функцию
- `k` – целочисленная переменная, которая считывает количество введённых в массив `arr[]` чисел (значение по умолчанию – 0)
- `i` – целочисленная переменная, которая считывает индексы элементов массива `arr[]` (шаг по массиву)
- `s` – символьная переменная, считывающая символ после каждого введённого числа в массиве `arr[]`

2. В остальных функциях:

- `a[]` – целочисленный массив размера не больше `N` (объявлен через `#define N 20`)
- `i` – целочисленная переменная, которая считывает индексы элементов массива `a[]` (шаг по массиву)
- `n` – целочисленная переменная, которая считывает количество введённых в массив `a[]` чисел (значение по умолчанию – 0)
- `fn` – целочисленная переменная, присваивающая значение индекса первого отрицательного элемента (функция `index_first_negative`)
- `ln` – целочисленная переменная, присваивающая значение индекса последнего отрицательного элемента (функция `index_last_negative`)
- `mb` – целочисленная переменная, присваивающая значение произведения элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего

отрицательного (не включая элемент) (функция *multi\_between\_negative*)

- *mba* – целочисленная переменная, присваивающая значение произведения элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент) (функция *multi\_before\_and\_after\_negative*)

Функции, реализованные в программе:

- Функция *index\_first\_negative(int a[], int n)* принимает на вход массив *a[]* и переменную *n* (длина массива), после чего с помощью цикла *for* получает индекс первого отрицательного элемента и возвращает его.
- Функция *index\_last\_negative(int a[], int n)* принимает на вход массив *a[]* и переменную *n* (длина массива), после чего с помощью цикла *for* получает индекс последнего отрицательного элемента и возвращает его.
- Функция *multi\_between\_negative(int a[], int n)* принимает на вход массив *a[]* и переменную *n* (длина массива), после чего с помощью цикла *for* находит произведение элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент) и возвращает его.
- Функция *multi\_before\_and\_after\_negative(int a[], int n)* принимает на вход массив *a[]* и переменную *n* (длина массива), после чего с помощью цикла *for* находит произведение элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент) и возвращает его.
- Функция *main()* принимает на вход значение *a* и массив *arr[]* длиной *k* ( $k \leq N$ ) (значения массива считываются с помощью цикла *for*). Затем с помощью оператора *switch* соответственно каждому значению *a* – 0-3 (если *a* принимает другие значения, то выводится

строка «Данные некорректны») выполняется одна из подзадач путём вызова функции и выводится полученное в функции значение.

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 -2 3 4 -2 5 -3 2 4	0	Вызывается функция <code>index_first_negative</code> , которая возвращает индекс первого отрицательного элемента.
2.	1 -2 3 4 -2 3 -2	5	Вызывается функция <code>index_last_negative</code> , которая возвращает индекс последнего отрицательного элемента.
3.	2 -3 2 1 -2 3 -2 1 2 3	36	Вызывается функция <code>multi_between_negative</code> , которая возвращает произведение элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент).
4.	3 4 3 -55 17 43 -31 4 -2 1 4 2	-192	Вызывается функция <code>multi_before_and_after_negative</code> , которая возвращает произведение элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент).
5.	4 3 2 5 3 -4	Данные некорректны	Поданное значение не совпало ни с одним из предложенных (0-3).

## **Выводы.**

Были изучены основные управляющие конструкции языка C такие, как условия, циклы и оператор `switch`, а также их применение для написания простейших программ.

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. Для обработки команд пользователя использовались оператор *switch* с блоками *case* для вывода данных и блоком *default* для сообщения о некорректности введенных данных, цикл `for` для ввода значений массива и его обработки. Кроме того, реализованы функции, выполняющие команды, в соответствии с условиями, требуемыми в задаче.



## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Kha\_lab1.cpp

```
#include <stdio.h>
#include <stdlib.h>
#define N 20

int index_first_negative(int a[], int n)
{
    int fn = 0;
    for (int i = n - 1; i >= 0; i--) {
        if (a[i] < 0) {
            fn = i;
        }
    }

    return fn;
}

int index_last_negative(int a[], int n)
{
    int ln = n - 1;
    for (int i = 0; i < n; i++) {
        if (a[i] < 0) {
            ln = i;
        }
    }

    return ln;
}

int multi_between_negative(int a[], int n)
{
    int mb = 1;
    for (int i = index_first_negative(a, n); i <
index_last_negative(a, n); i++) {
        mb *= a[i];
    }

    return mb;
}

int multi_before_and_after_negative(int a[], int n)
{
    int mba = 1;
    for (int i = 0; i < index_first_negative(a, n); i++) {
        mba *= a[i];
    }
    for (int i = index_last_negative(a, n); i < n; i++) {
        mba *= a[i];
    }

    return mba;
}
```

```

int main()
{
    char s;
    int arr[N], a, k = 0;
    scanf("%d", &a);
    for (int i = 0; i < N; i++) {
        scanf("%d%c", &arr[i], &s);
        k++;
        if (s == '\n')
            break;
    }

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

```