

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. 0382

Деткова А.С.

Преподаватель

Жангиров Т.Р.

САНКТ-ПЕТЕРБУРГ

2020

Цель работы.

Изучить основы языка Си (инициализация переменных, массивов и передача их в функции). Научиться использовать условный оператор (*if*), цикл *for*, оператор *switch*.

Задание.

Вариант №3.

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 100. Числа разделены пробелами. Строка заканчивается символом перевода строки.

В зависимости от значения, функция должна выводить следующее:

0: индекс первого нулевого элемента. (*index_first_zero*)

1: индекс последнего нулевого элемента. (*index_last_zero*)

2: Найти сумму модулей элементов массива, расположенных от первого нулевого элемента и до последнего. (*sum_between*)

3: Найти сумму модулей элементов массива, расположенных до первого нулевого элемента и после последнего. (*sum_before_and_after*)

иначе необходимо вывести строку "Данные некорректны".

Ошибкой в данном задании считается дублирование кода!

Подсказка: функция нахождения модуля числа находится в заголовочном файле *stdlib.h* стандартной библиотеки языка Си.

При выводе результата, не забудьте символ переноса строки.

Основные теоретические положения.

Был использован ***stdlib.h*** — заголовочный файл стандартной библиотеки языка Си, который содержит в себе функции, занимающиеся выделением памяти, контролем процесса выполнения программы, преобразованием типов и другие. Название «*stdlib*» расшифровывается как «standard library» (стандартная библиотека).

Функция *abs()* – нахождение модуля (абсолютного значения).

Выполнение работы.

На вход подается число *a*, которое в дальнейшем будет проверяться в операторе *switch*. Создается массив *arr* размером 100. Все введенные значения считываются и сохраняются в массив, размер имеющей значение части считается с помощью цикла *for*, в котором в переменную *arr_size* записывается размер массива, введенного с клавиатуры. Как только введенное значение не равно пробелу, прекращается ввод в массив и начинается проверка значения *a*. В зависимости от значения переменной *a*, выполняются следующие функции:

“0”: *int index_first_zero (int arr[], int n)* – на вход подается имя массива и его размер, а на выход – индекс первого элемента массива со значением “0”. Если такого элемента нет, то функция возвращает значение -1, в дальнейшем на экран выводится сообщение об ошибке.

“1”: *int index_last_zero (int arr[], int n)* - на вход подается имя массива и его размер, а на выход – индекс последнего элемента массива со значением “0”. Если такого элемента нет, то функция возвращает значение 100, в дальнейшем на экран выводится сообщение об ошибке.

“2”: *int sum_between (int arr[], int n)* - на вход подается имя массива и его размер, а на выход – сумма абсолютных значений между первым нулевым элементом и последним нулевым элементом. Если не был найден индекс первого нулевого и последнего нулевого элементов, то функция возвращает значение -1, в дальнейшем на экран выводится сообщение об ошибке. Если в массиве только один 0, то сумма считается равной нулю.

“3”: *int sum_before_and_after (int arr[], int n)* - на вход подается имя массива и его размер, а на выход – сумма абсолютных значений до первого нулевого элемента и после последнего нулевого элемента. Если не был найден индекс первого нулевого и последнего нулевого элементов, то функция возвращает значение -1, в дальнейшем на экран выводится сообщение об ошибке. Если в массиве только один 0, то сумма считается равной сумме абсолютных значений всех элементов.

Иначе, если переменная *a* не принимает ни одно из этих значений, то выводится сообщение об ошибке.

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 5 3 4 5 -6 7 -3	Данные некорректны	В массиве нет нулей, выводится сообщение об ошибке.
2.	0 3 4 0 5 6 7 -3 0 4 -10	2	Выводится индекс первого элемента массива со значением 0.
3.	1 2 -30 -3 4 5 6 3 4 6 -12	Данные некорректны	В массиве нет нулей, выводится сообщение об ошибке.
4.	1 2 3 0 3 4 -2 3 4 0 5 6	8	Выводится индекс последнего элемента массива со значением 0.
5.	2 3 4 5 6 7 -23 4 -12 34	Данные некорректны	В массиве нет нулей, выводится сообщение об ошибке.
6.	2 3 4 5 0 6 -2 3 5 7	0	В массиве один 0, индекс 1ого нулевого и последнего совпадают, сумма элементов между ними равна 0.
7.	2 3 4 0 5 -3 2 0 4 -10	10	Выводится сумма абсолютных значений, лежащих между первым и последним нулевыми значениями.
8.	3 -4 72 18 -3 5 6 7 9	Данные некорректны	В массиве нет нулей, выводится сообщение об ошибке.
9.	3 -2 -1 1 0 3 -1 1 2	11	В массиве один 0, индекс 1ого нулевого и последнего совпадают, сумма элементов до и после равна сумме всех элементов.
10.	3 4 -2 0 4 0 43 5 0 -1	7	Выводится сумма абсолютных значений, лежащих до первого и после последнего нулевого значения.
11.	9 4 6 0 3 0 0 2 3 5	Данные некорректны	Переменная <i>a</i> не приняла нужное значение, выводится сообщение об ошибке.

Выводы.

Были изучены основные управляющие конструкции языка Си, такие как: условный оператор *if*, оператор *switch*, цикл *for*. Исследована инициализация переменных и массивов, передача их в функции.

Разработана программа, выполняющая считывание с клавиатуры исходных данных в виде массива и команды пользователя. Программа находила, в зависимости от введенного значения команды пользователя, индекс первого или последнего нулевого элемента или сумму абсолютных значений элементов массива между первым и последними нулевыми элементами, или сумму абсолютных значений элементов массива до первого нулевого и после последнего нулевого элементов. Для решения задачи использовались функции, цикл *for*, оператор множественного выбора *switch*, оператор *if*, функция *abs()*.

Приложение А

Исходный код программы

Название файла: lb1.c

```
#include <stdio.h>
#include <stdlib.h>

int index_first_zero(int arr[], int n)
{
    int index, index_first = -1;
    for (index = 0; index < n; index++)
    {
        if (arr[index] == 0 && index_first == -1)
            index_first = index;
    }
    return index_first;
}

int index_last_zero(int arr[], int n)
{
    int index, index_last = 100;
    for (index = 0; index < n; index++)
    {
        if (arr[index] == 0)
            index_last = index;
    }
    return index_last;
}

int sum_between(int arr[], int n)
{
    int ind_first, ind_last, sum = 0, item;
    ind_first = index_first_zero(arr, n);
    ind_last = index_last_zero(arr, n);
    if (ind_first == -1 || ind_last == 100)
        return -1;
    for (item = ind_first; item <= ind_last; item++)
        sum = sum + abs(arr[item]);
    return sum;
}

int sum_before_and_after(int arr[], int n)
{
    int ind_first, ind_last, sum = 0, item;
    ind_first = index_first_zero(arr, n);
    ind_last = index_last_zero(arr, n);
    if (ind_first == -1 && ind_last == 100)
        return -1;
    for (item = 0; item <= ind_first; item++)
        sum = sum + abs(arr[item]);
    for (item = ind_last; item < n; item++)
        sum = sum + abs(arr[item]);
    return sum;
}
```

```

int main()
{
    int a, arr[100], arr_size, res;
    char sym = ' ';
    scanf("%d", &a);
    for (arr_size = 0; arr_size < 100 && sym == ' '; arr_size++)
    {
        scanf("%d%c", &arr[arr_size], &sym);
    }
    switch(a)
    {case 0:
        res = index_first_zero(arr, arr_size);
        if (res != -1)
            printf("%d\n", res);
        else
            printf("Данные некорректны\n");
        break;
    case 1:
        res = index_last_zero(arr, arr_size);
        if (res != 100)
            printf("%d\n", res);
        else
            printf("Данные некорректны\n");
        break;
    case 2:
        res = sum_between(arr, arr_size);
        if (res != -1)
            printf("%d\n", res);
        else
            printf("Данные некорректны\n");
        break;
    case 3:
        res = sum_before_and_after(arr, arr_size);
        if (res != -1)
            printf("%d\n", res);
        else
            printf("Данные некорректны\n");
        break;
    default:
        printf("Данные некорректны\n");
    }
    return 0;
}

```