

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. 1384

Усачева Д.В.

Преподаватель

Жангиров Т. Р.

Санкт-Петербург

2021

Цель работы.

Целью работы является ознакомление с управляющими конструкциями языка C на примере использующей их программы.

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.

Выполнение работы.

Для выполнения задачи были реализованы функции:

- 1) `index_first_zero(int arr[],int index)`-функция для определения индекса первого нулевого элемента. Она принимает на вход массив целых

чисел **arr**(от англ. *array* – массив) и целое число **index**, которое служит указателем на конкретный элемент массива. При помощи цикла *for* исследуем массив (размера не больше 100), начиная с нулевого элемента, используя условный оператор *if*, найдем индекс первого нулевого элемента и, записав его в переменную **n1**(от англ. *number 1-номер 1*), прервем цикл, применив оператора *break*.

2) **index_last_zero(int arr[],int index)**-функция для определения индекса последнего нулевого элемента. Аналогично первой функции она принимает на вход массив целых чисел **arr**(от англ. *array* – массив) и целое число **index**, которое служит указателем на конкретный элемент массива и хранит в себе информацию о количестве введенных элементов массива. При помощи цикла *for* исследуем массив (размера не больше 100), начиная с последнего элемента (**arr[index]**), используя условный оператор *if*, найдем индекс последнего нулевого элемента и, записав его в переменную **nn**(от англ. *number n –номер n-ого нулевого элемента*), прервем цикл, при помощи оператора *break*.

3) **sum_between(int arr[],int index)**- функция для поиска суммы модулей элементов массива, расположенных от первого нулевого элемента и до последнего. Она принимает на вход массив целых чисел **arr**(от англ. *array* – массив) и целое число **index**, которое служит указателем на конкретный элемент массива и хранит в себе информацию о количестве введенных элементов массива. В целях избегания дублирования кода при помощи функций (1) и (2) найдем первый и последний нулевые элементы массива и сохраним их индексы в переменные **index1** и **indexn** соответственно. Используя цикл *for* суммируем модули элементов массива, от **arr[index1]** до **arr[indexn]**. Результат сложения запишем в предварительно заданную переменную **sum_b**(от англ. *sum between- сумма между*), изначально

равную 0.

- 4) *sum_before_and_after(int arr[],int index)*- функция для поиска суммы модулей элементов массива, расположенных до первого нулевого элемента и после последнего. Она принимает на вход массив целых чисел **arr**(от англ. *array* – массив) и целое число **index**, которое служит указателем на конкретный элемент массива и хранит в себе информацию о количестве введенных элементов массива. В целях избегания дублирования кода при помощи функций (1) и (2) найдем первый и последний нулевые элементы массива и сохраним их индексы в переменные **index1** и **indexn** соответственно. Дважды используя цикл *for*, суммируем модули элементов массива сначала расположенные до первого нулевого элемента, далее - после последнего. Результат сложения запишем в предварительно заданную переменную **sum_b_a**(от англ. *sum before and after* - сумма до и после), изначально равную 0.
- 5) В основной программе в первую очередь было введено число, сохраненное в переменную **number**(от англ. *number* – число). Его роль- задание алгоритма для дальнейшей работы программы (целое число от 0 до 3). Затем был создан массив целых чисел **arr** (от англ. *array* – массив), в котором будут храниться введенные из входных данных числа. Так как указано лишь, что размер массива не больше 100, точное количество его элементов неизвестно. Но мы знаем, что строка заканчивается символом перевода строки(**\n**). Используя цикл *do-while*, остановим ввод данных, когда **f**, ранее являющийся разделителем « » вводимых чисел (от англ. *flag-флаг*) станет равен «**\n**». Затем используется оператор *switch* для выбора необходимой нам функции.

Разработанный программный код см. в приложении 1

Результаты тестирования см. в приложении 2

Выводы

Была освоена работа с управляющими конструкциями на языке С. Были изучены основные управляющие конструкции языка: *for*, *if*, *do-while*, *switch-case*.

Разработана программа, выполняющая считывание с клавиатуры команды пользователя и исходного массива. Для обработки команд пользователя использовался оператор *switch-case*. В зависимости от введённой команды программа находила: индекс первого нулевого элемента массива, индекс последнего нулевого элемента массива, сумма модулей элементов массива, расположенных от первого нулевого элемента и до последнего, сумма модулей элементов массива, расположенных до первого нулевого элемента и после последнего. В случае некорректного ввода номера команды программа выводит текст «Данные некорректны» и завершает работу.

ПРИЛОЖЕНИЕ 1

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Usacheva_Daria_lb1

```
#include <stdio.h>
#include <stdlib.h>
int index_first_zero(int arr[],int index){
    int n1=0;
    for(index=0;index<=100;index++){
        if(arr[index]==0){
            n1=index;
            break;
        }
    }
    return n1;
}
int index_last_zero(int arr[],int index){
    int nn=0;
    for(;index>=0;index--){
        if(arr[index]==0){
            nn=index;
            break;
        }
    }
    return nn;
}
int sum_between(int arr[],int index){
    int sum_b=0;
    int index1=index_first_zero(arr,index);
    int indexn=index_last_zero(arr,index);
    for (;index1!=indexn;index1++){
        sum_b=sum_b+abs(arr[index1]);
    }
    return sum_b;
}
int sum_before_and_after(int arr[],int index){
```

```

int sum_b_a=0;
int index1=index_first_zero(arr,index);
int indexn=index_last_zero(arr,index);
for (int index0=0;index1>=index0;index0++)
    sum_b_a=sum_b_a+abs(arr[index0]);
for (;indexn<=index;indexn++)
    sum_b_a=sum_b_a+abs(arr[indexn]);
return sum_b_a;
}

int main(){
    int number ;
    scanf("%d",&number);
    char f;
    int arr[100],index=0;
    do{
        scanf("%d%c", &arr[index],&f);
        index++;
    } while (f!='\n');
    --index;
    switch (number){
        case 0:
            printf("%d\n", index_first_zero(arr,index));
            break;
        case 1:
            printf("%d\n",index_last_zero(arr,index));
            break;
        case 2:
            printf("%d\n",sum_between(arr,index));
            break;
        case 3:
            printf("%d\n",sum_before_and_after(arr,index));
            break;
        default:
            puts("Данные некорректны\n");
            break;
    }
}

```

```
    }  
    return 0;  
}
```


ПРИЛОЖЕНИЕ 2

ТЕСТИРОВАНИЕ

Таблица 2.1 - Пример тестовых случаев

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 107 61 59 -48 90 122 -101 -90 44 140 0 -136 16 -7 -26 - 79 150 -20 0 -124 -19 89 101 -98 -27 -67 84 -88 96 -21 -89 63	10	Индекс первого нулевого элемента массива равен 10
2.	23 0 56 -116 8 -36 146 -6 107 147 30 -135 73 146 -45 - 103 -85 6 120 -33 -98 -150 51 126 53	20	Индекс последнего нулевого элемента массива равен 20
3.	2 -116 -76 -134 -21 -147 102 145 36 -136 107 34 -89 136 34 -35 67 18 -12 -32 88 0 -30 -51 -79 -10 -147 -102 98 46 - 135 -48 -53 -44 -88 135 43 0 -38 -38 -36 -3 82 80 49 -29 147 131 71 103 0 -59 21 -113 -148 -138 81 121 37 -29 143	1916	Сумма модулей элементов массива, расположенных от первого нулевого элемента и до последнего равна 1916
4.	3 48 32 133 -148 23 -42 45 - 98 -34 59 120 0 -94 59 39 - 124 -50 8 -127 -130 0 13 70 - 102 7 112 75 80 -117 -115 69 -121	1663	Сумма модулей элементов массива, расположенных до первого нулевого элемента и после последнего равна 1663

5.	3 -131 -18 -83 -96 13 42 0 108 -68 -8 -137 -6 -122 16 106 -50 -22 12 79 -73 -87 -7 82 -58 -120 -11 91 -142 46 - 98 93 40 75 -27 123 44 -23 0 44 -104 -79 -25 -26 38 -149 14 -114 -42 63 77 -38 -56 -98 -17 -57 -46 -60 -54 -31 0 - 106 58 -4 132 -104 -77 -28 139 9 141 41 86 -101 147 129 -110 92 -18 -56 45 -81 0 -94 59 39 -124 -50 8 -127 - 130 0 13 70 -102 7 112 75 80 -117 -115 69 -121	1264	Сумма модулей элементов массива, расположенных до первого нулевого элемента и после последнего равна 1264
----	---	------	---