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: Управляющие конструкции языка Си.

Студент гр. 0382 _____ Санников В.А.

Преподаватель _____ Чайка К.В.

Санкт-Петербург
2020

Цель работы.

Изучить основные управляющие конструкции языка Си: Условия, циклы и оператор switch.

Задание.

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".
```

Основные теоретические положения.

•Функции библиотеки stdio.h:

◦printf()–функция выводит на консоль значине аргумета

◦scanf()–функция ввода данных из консоли

•Функция библиотеки stdlib.h:

◦abs()–функция получения модуля числа

•Циклы:

◦while(){}–каждая итерация проверяет, выполняется ли условие в круглых скобках, если оно верно, то выполняется код в фигурных скобках, а если неверно, то происходит выход из цикла

◦for(){<переменная>; <условие>; <выражение_1>}–код в теле цикла будет исполнятся до тех пор, пока объявленная в цикле переменная будет удовлетворять условию цикла, выражение_1 каким-либо способом меняет значение этой переменной

•Операторы:◦if(){} ... else{}–если выполняется условия, указанное в круглых скобках, то выполняется код в фигурных скобках после if, иначе–в фигурных скобках после else(else не является обязательной частью конструкции)

◦switch(<переменная>){casex:... break; ... default:...break;}–от значения переменной в круглых скобках зависит, какой кейс будет выполняться (например, если переменная имеет значение x–выполнится casex). Если же не будет кейса с таким значением, то выполнится код из блока default

•Функции:◦<тип_функции> имя_функции(<аргумент_1>, ... , <аргумент_n>) {}–при вызове данной функции в главной(main) функции выполняется код в фигурных скобках, а затем возвращает значение оператором return(если тип функции не void)

Выполнение работы.

Ход работы:

В начале программы подключаем библиотеки:

•stdio.h–используется для подключения ввода-вывода (printf(), scanf())

•stdlib.h–используются для доступа к функции abs(), которая позволяет вычислить модуль числа.

Затем объявим переменные:

массив numb[100] — будет хранить входной массив целых чисел

prob — переменная типа char, которая будет хранить символ, разделяющий элементы массива

length — переменная типа int, это длинна массива, передается для контроля выхода из цикла.

Для начала считываем данные с помощью оператора scanf. Сначала считывается переменная symbol типа int, определяющая какая из функций (index_first_zero,index_last_zero,sum_between,sum_before_and_after)будет вызвана. Далее с помощью while считываем элементы массива с клавиатуры так, чтобы они были через пробел и не превышали 100 штук.

1) Чтобы найти индекс первого нулевого элемента в массиве воспользуемся циклом for: в его условии зададим, чтобы он перебирал индексы элементов массива и остановился на нулевом элементе. С помощью оператора return возвращаем индекс нужного нам элемента.

2) Для нахождения индекса последнего нулевого элемента массива снова воспользуемся циклом for. В начале функции объявим переменную thelast типа int. Далее перебираем элементы массива с помощью for и присваиваем индекс нулевых элементов переменной i в цикле for с помощью оператора if. С помощью оператора return возвращаем переменную thelast.

3) А теперь найдем сумму модулей элементов массива, которые находятся между первым и последним нулевыми элементами массива. Объявим переменные sum1 и sum2 типа int, sum 2 будет выполнять роль сумматора т. е суммировать модули элементов массива (про sum1 позже). Для того, чтобы дойти до первого нулевого элемента в массиве воспользуемся циклом for, далее используем цикл while (он будет продолжаться, пока не закончатся все элементы массива), внутри цикла while используем снова for, только для элементов, идущих после первого нулевого элемента массива, при этом начинаем суммировать модули элементов, пока снова не дойдем до нулевого элемента. Далее после for внутри while создаем условие с помощью if: оно нужно для того, чтобы при нахождении нулевого элемента, sum1 присвоить sum2. Данное условие позволяет нам посчитать сумму модулей элементов именно до последнего нулевого элемента. К примеру(без if), если

бы в массиве было 3 нуля, то сумматор посчитал бы сумму модулей до 2 нулевого элемента. С помощью оператора return возвращаем переменную sum1.

4) Чтобы найти сумму модулей элементов массива до первого нулевого элемента и после последнего, сначала объявим переменные sum1 и sum2 типа int, которые будут выполнять роль сумматоров. Далее с помощью цикла for суммируем модули элементов массива с помощью sum1, пока не дойдем до первого нулевого элемента. Потом, используя цикл for, суммируем в sum2 элементы, находящиеся после нулевого, при этом проверяя условие в цикле: не нулевой ли этот элемент. Если данный элемент не нулевой, то суммируем его в sum2, а если нулевой, то обнуляем sum2. Это условие сделано с помощью оператора if...else для того, чтобы просуммировать элементы именно после последнего нулевого элемента. С помощью оператора return возвращаем sum1 + sum2.

Чтобы выполнить один из пунктов (1-4), внедряем в нашу программу оператор switch. Если данные введены неправильно, выводим «Данные некорректны».

В конце программы: return 0.

Краткий алгоритм: ввод данных с клавиатуры(scanf) → заходим в оператор switch, выбирается ветка действий в зависимости от введенных данных → срабатывает функция → получаем ответ, либо сообщение о некорректности введенных данных.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	`0 -21 10 0 -23 -7 -15 -14 8 -9 10 -13 -14 -27 0 -7 12 -18`	2	Первый нулевой элемент с индексом 2
2.	1 2 4 -5 0 -56 0 34 3 0	8	Последний нулевой элемент с индексом 8
3.	2 3 0 4 -45 4 0 45 0	94	Сумма модулей элементов между первым нулевым и последним
4.	3 4 5 0 -45 89 2 0 4	16	Сумма модулей элементов перед первым нулевым и после последнего

Выводы.

С трудом, но я познал и изучил азы основных управляющих конструкций языка Си (switch, for, if...else, while, do...while).

Разработана программа, выполняющая краткий алгоритм: ввод данных с клавиатуры(scanf) → заходим в оператор switch, выбирается ветка действий в зависимости от введенных данных → срабатывает функция → получаем ответ, либо сообщение о некорректности введенных данных.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла:

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

int index_first_zero(int fnumb[], int length){
    int i;
    for (i=0; i<length && fnumb[i]!=0; i++){
    }
    return i;
}

int index_last_zero(int fnumb[], int length){
    int i, thelast=0;
    for (i=0; i<length; i++){
        if (fnumb[i]==0) thelast = i;
    }
    return thelast;
}

int sum_between(int fnumb[], int length){
    int i, sum1=0, sum2=0;
    for (i=0;fnumb[i]!=0;i++){
    }
    while (i<length){
        for (i++; i<length && fnumb[i]!=0; i++){
            sum2+=abs(fnumb[i]);
        }
            if (i<length && fnumb[i]==0)
            sum1=sum2;
    }
    return sum1;
}

int sum_before_and_after(int fnumb[], int length){
    int i, sum1=0, sum2=0;
    for (i=0; fnumb[i]!=0; i++){
        sum1+=abs(fnumb[i]);
    }
    for (i++; i<length; i++){
        if (fnumb[i]!=0){
            sum2+=abs(fnumb[i]);
        }
        else{
            sum2=0;
        }
    }
    return sum1 + sum2;
}

int main(){
    int numb[100], length=0, symbol;
```

```
    char prob = ' ';

    scanf("%d", &symbol);

    while (length < 100 && prob == ' '){
        scanf("%d%c", &numb[length], &prob);
        length++;
    }
    switch(symbol){
case 0: printf("%d", index_first_zero(numb,length));
    break;
case 1: printf("%d", index_last_zero(numb,length));
    break;
case 2: printf("%d", sum_between(numb,length));
    break;
case 3: printf("%d", sum_before_and_after(numb,length));
    break;
default:printf ("Данные некорректны");
    break;
}
    return 0;
}
```