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 _____ Шаврин А.П

Преподаватель _____ Чайка К.В.

Санкт-Петербург
2021

Цель работы.

Изучить основные управляющие конструкции языка Си, функции, типы данных, циклы, условные операторы и получить навыки работы с ними.

Задание.

3 вариант.

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.

Цикл for:

for (<Инициализация>; <Условие>; <Модификация>){

<БлокОпераций>;

}

Цикл while:

while(<Условие>){

<БлокОпераций>;

}

Условный оператор if-else:

if (<Условие>){

<БлокОпераций1>;

}

else{

<БлокОпераций2>;

}

Условный оператор switch:

```
switch(<Условие>){
    case(<Значение 1>):
        <БлокОпераций 1>;
        break;
    ...
    case(<Значение n>):
        <БлокОпераций n>;
        break;
    default:
        <БлокОперацийПоУмолчанию>;
}
```

Выполнение работы.

С помощью *#include* в программу включаются файлы *<stdio.h>* и *<stdlib.h>*.

Затем с помощью *#define* определяется значение по умолчанию для максимальной длины массива (*N*) равное 100.

В функции *main()* инициализируются: массив *arr[N]*, целочисленные переменные *x* и *n*. После в переменную *x* с помощью функции стандартной библиотеки *scanf()* записывается первое введеное пользователем значение.

Для записи массива создается функция *readArr()*, которая принимает в качестве аргументов массив *arr[]* и его максимальную длину *len*, а возвращает его действительную длину. В функции создается переменная *c* типа *char,* для отлавливания знака переноса строки *'\n',* и *n* типа *int,* для записи действительной длины массива. В функции с помощью цикла *while* и условного оператора *if*, происходит заполнение массива и запись его действительной длинны. После в функции *main* в переменную *n* записывается значение, возвращаемое функцией *readArr()*.

Поскольку программа должна реализовать 4 различных действия, в зависимости от первого введенного пользователем значения, создается 4 функции под каждое действие.

Для реализации первого действия создается функция *index_first_zero()*, принимающая на вход массив целочисленных значений *arr[]* и его длину *len*, а возвращающая целое число. В функции создается целочисленная переменная *n*, для записи индекса первого нуля. Для прохода по массиву и поиску необходимого значения был написан цикл *while* с условным оператором *if*.

Для реализации второго действия создается функция *index_last_zero()*, принимающая на вход массив целочисленных значений *arr[]* и его длину *len*, а возвращающая целое число. В функции создается целочисленная переменная *n*, для записи индекса последнего нуля. Для прохода по массиву в обратном порядке и поиску необходимого значения был написан цикл *for* с условным оператором *if*.

Для реализации третьего действия создается функция *sum_between()*, принимающая на вход массив целочисленных значений *arr[]*, его длину *len*, индекс первого нуля *index_first_zero* и индекс последнего нуля в массиве *index_last_zero*, а возвращающая сумму значений между этими нулями. Для записи суммы создается переменная *sum* типа *int*. Для прохода по массиву от первого нуля до последнего был написан цикл *for*.

Для реализации четвертого действия создается функция *sum_ before_and_after()*, принимающая на вход массив целочисленных значений *arr[]*, его длину *len*, индекс первого нуля *index_first_zero* и индекс последнего нуля в массиве *index_last_zero*, а возвращающая сумму значений от начала массива до первого нуля и от последнего нуля и до конца. Для записи суммы создается переменная *sum* типа *int.* Для прохода по массиву был написан цикл *for*, а для пропуска значений от первого нуля и до второго нуля был использован условный оператор *if* и оператор *continue*.

Для реализации выполнения действия соответствующего первому значению введенному пользователем создается функция user_choice(), которая принимает на вход первое значение введенное пользователем (*int x),* массив целочисленных значений *arr[],* его длину *len.* В функции создаются целочисленные переменные переменные ifz и ilz (сокращения от *index_first_zero* и *index_last_zero*), в которые записываются значения,

возвращаемые соответствующими названиям переменным, функций. При помощи условного оператора *switch* реализуется выполнение функций, соответствующих определенным значениям переменной *x*. В случае если пользователь ввел не предусмотренное заданием значение, выполняется действие по умолчанию (*default*).

В функции *main* выполняется вызов функции *user_choice(),* с передачей соответствующих аргументов.

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 -3 3 0 4 -5 4 0 7 -9 0 4 0 4 -4 -5	2	Функция по поиску индекса первого нуля в массиве выполнилась корректно.
2.	1 -3 3 0 4 -5 4 0 7 -9 0 4 0 4 -4 -5	11	Функция по поиску индекса последнего нуля в массиве выполнилась корректно.
3.	2 -3 3 0 4 -5 4 0 7 -9 0 4 0 4 -4 -5	33	Функция по вычислению суммы значений в массиве от первого нуля до последнего выполнилась корректно.
4.	3 -3 3 0 4 -5 4 0 7 -9 0 4 0 4 -4 -5	19	Функция по вычислению суммы значений в массиве до первого нуля и после последнего выполнилась корректно.
5.	4 -3 3 0 4 -5 4 0 7 -9 0 4 0 4 -4 -5	Данные некорректны	Значение пользователя было не корректно и выполнилось действие по умолчанию.

Выводы.

Я изучил основные управляющие конструкции языка Си, функции, типы данных, циклы, условные операторы и получил навыки работы с ними. Мною была разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя, с помощью функции стандартной библиотеки *scanf()*. Для обработки команд пользователя использовался условный оператор *if,* оператор *switch,* цикл *while*, *for* и оператор *break*.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Lab1.c

```
#include <stdio.h>
#include <stdlib.h>
#define N 100

int readArr(int arr[], int len){
	char c;
	int n=0;
	while(n < len){
		scanf("%d%c", &arr[n], &c);
		n++;
		if(c == '\n')
			break;
	}
	return n;
}

int index_first_zero(int arr[], int len){
	int n=0;
	while(n < len){
		if(arr[n] == 0)
			return n;
		n++;
	}
}

int index_last_zero(int arr[], int len){
	for(int i=len-1; i>=0; i--){
		if(arr[i] == 0)
```

```
                return i;
        }
    }

    int sum_between(int arr[], int len, int index_first_zero, int  index_last_zero){
        int sum=0;
        for(int i=index_first_zero+1; i != index_last_zero; i++){
            sum += abs(arr[i]);
        }
        return sum;
    }

    int sum_before_and_after(int arr[], int len, int index_first_zero, int index_last_zero){
        int sum=0;
        for(int i=0; i<len; i++){
            if((i >= index_first_zero) && (i <= index_last_zero))
                continue;
            sum += abs(arr[i]);
        }
        return sum;
    }

    void user_choice(int x, int arr[], int len){
        int ifz = index_first_zero(arr, len);
        int ilz = index_last_zero(arr, len);
        switch(x){
            case 0:
                printf("%d\n", ifz);
                break;
            case 1:
```

```
            printf("%d\n", ilz);
            break;
        case 2:
            printf("%d\n", sum_between(arr, len, ifz, ilz));
            break;
        case 3:
            printf("%d\n", sum_before_and_after(arr, len, ifz, ilz));
            break;
        default:
            puts("Данные некорректны\n");
    }
}

int main(){
    int arr[N];
    int x;
    int n;
    scanf("%d", &x);
    n = readArr(arr, N);
    user_choice(x, arr, n);
}
```