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 _____ Чернякова В.А.

Преподаватель _____ Чайка К.В.

Санкт-Петербург
2021

Цель работы.

Освоение работы с управляющими конструкциями языка С на примере использующей их программы. Научиться работать с условным оператором, циклами и оператором множественного выбора, 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 : максимальное число в массиве. (max)

1 : минимальное число в массиве. (min)

2 : разницу между максимальным и минимальным элементом. (diff)

3 : сумму элементов массива, расположенных до первого минимального элемента. (sum)

иначе необходимо вывести строку "Данные некорректны".

Выполнение работы.

В главной функции *int main ()* объявляем целочисленные типы данных: динамический массив *arr,* в котором будут храниться вводимые с клавиатуры значения, точное количество которых неизвестно, переменную *ind = 0*, которая отвечает за длину массива, и переменную *n = 0,* в которой будет храниться значение для оператора *switch.*

Объявляем символьную переменную *symb,* в которой будет храниться знак, следующий за числом: пробел или перенос строки.

С помощью функции *scanf* считываем значение в переменную *n*.

С помощью цикла *while* заполняем наш массив *arr* до тех пор, пока значение, считанное в символьную переменную *symb* не станет равным символу переноса строки - *\n*. Внутри цикла с помощью функции *scanf*

считываем значение *arr[ind]* и *symb,* затем увеличиваем значение *ind* на единицу *ind*++.

Вызываем оператор *switch*, в который передаем значение переменной *n*. В зависимости от переданного значения переменной *n* описываем блоки оператора.

Блок *case 0:* с помощью функции *printf* выводится результат работы функции *max()*, принимающей на вход массив *arr* и длину массива *ind* и возвращающей значение максимального числа в массиве. Прерываем блок с помощью оператора *break*.

Блок *case 1:* с помощью функции *printf* выводится результат работы функции *min(),* принимающей на вход массив *arr* и длину массива *ind* и возвращающей значение минимального числа в массиве. Прерываем блок с помощью оператора *break*.

Блок *case 2:* с помощью функции *printf* выводится результат работы функции *diff(),* принимающей на вход массив *arr* и длину массива *ind* и возвращающей значение разницы между максимальным и минимальным элементом этого массива. Прерываем блок с помощью оператора *break*.

Блок *case 3:* с помощью функции *printf* выводится результат работы функции *sum(),* принимающей на вход массив *arr* и длину массива *ind* и возвращающей значение суммы элементов массива, расположенных до первого минимального элемента этого массива. Прерываем блок с помощью оператора *break*.

Блок *default:* данный блок срабатывает в том случае, если ни одно из значений не совпало, тогда с помощью функции *printf* выводится *«Данные некорректны»*. Прерываем блок с помощью оператора *break*.

При выводе результата с помощью функции *printf (“\n”)* не забываем использовать символ переноса строки (примечание в условии к заданию лабораторной работы).

Функции:

Функция *int max(int array[], int index)* принимает на вход в качестве аргументов массив целых чисел *array[]* и целочисленную переменную *index,* в которой хранится значение длинны массива. В функции объявляется целочисленная переменная *maximum = array[0]*, значение которой изначально равно нулевому элементу массива. С помощью цикла *for (int i = 0; i < index; i++)* проходимся по каждому элементу массива. В теле цикла используем условный оператор *if (array[i] > maximum),* который сравнивает значение элемента массива и максимума. Если условие выполняется, то переменной *maximum* присваивается значение *array[i].* По окончанию цикла функция возвращает значение максимального элемента массива *return maximum*.

Функция *int min(int array[], int index)* принимает на вход в качестве аргументов массив целых чисел *array[]* и целочисленную переменную *index*, в которой хранится значение длинны массива. В функции объявляется целочисленная переменная *minimum = 101*, значение которой изначально равно 101. С помощью цикла *for (int i = 0; i < index; i++)* проходимся по каждому элементу массива. В теле цикла используем условный оператор *if (array[i] < minimum)*, который сравнивает значение элемента массива и минимума. Если условие выполняется, то переменной *minimum* присваивается значение *array[i]*. По окончанию цикла функция возвращает значение минимального элемента массива *return minimum*.

Функция *int diff(int array[], int index)* принимает на вход в качестве аргументов массив целых чисел *array[]* и целочисленную переменную *index*, в которой хранится значение длинны массива. В функции объявляется целочисленная переменная *difference*. Этой переменной присваивается значение разницы между максимальным элементом массива, пришедшего на вход функции, и минимальным элементом массива, пришедшего на вход этой же функции, *difference = max(array, index) - min(array, index)*. Высчитывание значения происходит с использованием функций, описанных раннее. Функция возвращает числовое значение разницы между самым большим и самым маленьким элементом массива *return difference*.

Функция *int sum(int array[], int index)* принимает на вход в качестве аргументов массив целых чисел *array[]* и целочисленную переменную *index*, в которой хранится значение длинны массива. В функции объявляются целочисленные переменные *summa = 0*, в которой будут складываться значения, и *i = 0*, отвечающая за индекс элемента массива. С помощью цикла *while (array[i] != min(array, index))* перебираем элементы массива до тех пор, пока i-тый элемент массива не станет равен первому минимальному элементу этого массива(значение минимального элемента высчитывается с помощью функции *min()*, описанной раннее). В теле цикла увеличиваем значение переменной *summa* на значение i-того элемента массива *summa = summa + array[i]*. Для продвижения по массиву внутри цикла увеличиваем значение переменной *i* на единицу *i++*. По завершению цикла функция возвращает значение суммы элементов массива, расположенных до первого минимального элемента этого массива *return summa*.

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 -21 10 0 -23 -7 -15 -14 8 -9 10 -13 -14 -27 0 -7 12 18	18	Вызывается блок с функцией *int max(),* которая возвращает значение максимального элемента массива *arr*.
2.	1 -8 -23 -30 -11 -28 15 -20 -24 -27 5 -13 5 21 -5 16 30 -12 15 -14 -28 -27 -11 -5 4 29 -5	-30	Вызывается блок с функцией *int min(),* которая возвращает значение минимального элемента массива *arr*.
3.	2 1 16 2 -18 -22 15 -3 13 0	50	Вызывается блок с функцией *int diff(),* которая

	-6 1 9 24 1 -18 15 28 20 -17 16 -11		возвращает значение разницы между максимальным элементом массива *arr* и его минимальным элементом.
4.	3 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 -26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9	0	Вызывается блок с функцией *int sum(),* которая возвращает значение суммы элементов массива *arr,* расположенных до первого минимального элемента этого же массива.
5.	12 -5 -3 -5 -8 3 -9 -3	Данные некорректны	Вызывается блок *default*, так как ни одно из значений ни совпало.

Выводы.

Я освоила работу с управляющими конструкциями языка С на примере программы, которую написала. Научилась работать с условным оператором, циклами и оператором множественного выбора, switch.

Были изучены основные управляющие конструкции языка: оператор *if, switch* и циклы *for, while.*

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. Для обработки команд пользователя использовались условные операторы *if*, циклы *for* и *while,* а также оператор *switch* с четырьмя блоками *case* и блоком *default,* сообщающим о некорректности веденных данных пользователем. Были написаны отдельные функции, которые выполняли команды, требуемые условием определенного блока.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Chernyakova_Valeria_lb1/main.c

```
#include <stdio.h>

int max(int array[], int index){
    int maximum = array[0];
    for (int i = 0; i < index; i++)
        if (array[i] > maximum)
        maximum = array[i];
    return maximum;
}

int min(int array[], int index){
    int minimum = 101;
    for (int i = 0; i < index; i++)
        if (array[i] <= minimum)
        minimum = array[i];
    return minimum;
}

int diff(int array[], int index){
    int difference;
    difference = max(array, index) - min(array, index);
    return difference;
}

int sum(int array[], int index){
    int summa = 0, i = 0;
    while (array[i] != min(array, index)){
        summa = summa + array[i];
        i++;
    }
    return summa;
}

int main(){
    int *arr, ind = 0, n = 0;
    char symb;
    scanf ("%d", &n);
    while (symb != '\n'){
        scanf ("%d%c", &arr[ind], &symb);
        ind++;
    }
    switch (n){
        case 0:
            printf ("%d\n", max(arr, ind));
            break;
        case 1:
            printf ("%d\n", min(arr, ind));
            break;
        case 2:
            printf ("%d\n", diff(arr, ind));
            break;
```

```
            case 3:
                printf ("%d\n", sum(arr, ind));
                break;
            default:
                printf ("Данные некорректны\n");
                break;
        }
    }
```