

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 C 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 длины 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;* условный оператор *if (array[i] > maximum)*, который сравнивает значение элемента массива и максимума. Если условие выполняется, то переменной 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 длины 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* минимума. Если условие выполняется, то переменной *minimum* присваивается значение *array[i]*. По окончании 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 длины 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)*. Вычитывание значения происходит с использованием функций, описанных ранее. 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 длины массива. В функции объявляются целочисленные переменные *summa = 0*, в которой будут складываться значения, и *i = 0*, отвечающая за индекс элемента массива. С помощью цикла пока *i*-тый элемент массива не станет равен первому минимальному элементу этого массива (значение минимального элемента высчитывается с помощью функции *min()*, описанной ранее). В теле цикла увеличиваем значение переменной *summa* на значение *i*-того элемента массива *summa = summa +* 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.			Вызывается блок с функцией <i>int max()</i> , которая возвращает значение максимального элемента массива <i>arr</i> .
2.			Вызывается блок с функцией <i>int min()</i> , которая возвращает значение минимального элемента массива <i>arr</i> .
3.			Вызывается блок с функцией <i>int diff()</i> , которая

			возвращает значение разницы между максимальным элементом массива <i>arr</i> и его минимальным элементом.
			Вызывается блок с функцией <i>int sum()</i> , которая возвращает значение суммы элементов массива <i>arr</i> , расположенных до первого минимального элемента этого же массива.
		Данные некорректны	Вызывается блок <i>default</i> , так как ни одно из значений не совпало.

Выводы.

Я освоила работу с управляющими конструкциями языка C 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*

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*, сообщаящим о некорректности введенных 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;  
    }  
}
```