

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

Макки К.Ю

Преподаватель

Чайка К.В

Санкт-Петербург

2021

Цель работы.

Научиться использовать управляющие конструкции языка Си, такие как условия, циклы, оператор switch

Задание.

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.

Необходимо написать программу, которая может выполнять 4 разных функции с массивом номеров, но на выбор пользователя. В этом массиве есть 2 части, выбор функции в виде одного числа и сам массив размером не больше 100 чисел. Все числа должны быть разделены пробелом, а символ перевода строки обозначает конец ввода пользователя. Программа начинается с функции `arr_imp(array input)`. Это функция принимает один 1 аргумент, `arr[]` где будет храниться введенные данные. Мы объявляем 2 переменные `sep(separator)` и `n(number)`. В переменной `sep` вида `char` будет храниться символ разделения между числами массива, а в `n` будет храниться количество чисел в массиве в виде `int`. Потом пишем цикл `for` который принимает 3 условия. Первое, `int i=0` который будет хранить в себе количество раз цикл повторяется, второй `i< 100` который говорит программе

повторять цикл пока `i` меньше 100, последний `i++` который прибавляет значение `i` каждый раз цикл повторяется. После `for` у нас функция `scanf` которая принимает данные от пользователя и ставит их в обозначенные переменные. После приема данных мы прибавляем `n` чтобы к концу цикла узнать количество раз он повторился и назначит это число как размер массива. Мы используем оператор `if` с условием (`ser == '\n'`) в случае это условие верное цикл сразу заканчивается. В конце функция `arr_inr` возвращает нам число `n` который будет хранить точный размер массива.

Вторая функция это `max_num(maximum number)` которая принимает 2 аргумента первый это массив `arr` и размер массива `n`. В функции мы объявляем переменную `max = arr[1]`, первое число в массиве. Потом используем цикл `for` который будет проходить через каждое число этого массива и сравнивать его с `max` если это число больше тогда `max` будет равен этому числу. В цикле `for` мы ставим индекс `i = 1` потому что первое число введенных данных не часть массива. Функция возвращает переменную `max`.

Третья функция это `min_num (минимальный number)` это функция тоже принимает 2 параметра такие же как прошлая. Это Функция находит самый маленький номер в массиве. Мы создаём переменную `min = arr[1]` первый элемент массива. Используя цикл `for` с такими же условиями как в прошлой функции. Внутри цикла `for` мы используем `if` с условием если число массива меньше переменной `min` этот элемент становится минимальным элементом.

Четвертая функция `diff_num(difference number)` в этой функции начинаем с объявления переменной `max` в которой вызываем функцию `max_num(arr,n)`. Это переменная будет равна максимальному числу массива. Потом делаем то же самое с переменной `min = min_num(arr,n)`, но с минимальным номером массива. Третья переменная `diff` будет равна разности максимального - минимального числа. И в конце функция возвращает эту переменную `diff`.

Пятая функция `sum_num(sum numbers)` должна вывести сумму всех чисел до минимального числа массива. Функция принимает те же самые 2 параметра как предыдущие функции. Начинаем и создаём переменную `sum=0` эта переменная будет хранить в себе сумму чисел. Ещё мы создаём переменную `min = min_num(arr,n)` а этой переменной мы вызываем функцию чтобы потом сравнивать числа массива с минимальным числом и узнать когда заканчивать цикл. В цикле, мы используем цикл `for` в котором лежит `if` которая, сравнивает числа с минимальным числом и когда числа равны цикл заканчивается если условие не верное число массива с индексом `i` добавляется в `sum` и становится им. Эта функция возвращает нам переменную `sum`.

Последняя функция `main()` и самая главная. В этой функции мы объявляем переменную `arr[100]` массив с максимальным размером 100 элементов типа `int`. После этого нам нужно объявить переменную `n` которая равна вызову функции `arr_inr` эта функция возвращает длину этого массива. Но главная функция её вызова это дать возможность пользователю ввести числа в массив. Потом мы создаём `switch` который сравнивает элемент массива с индексом 0 с разными `case`. Если `arr[0]==0` функция печатает на экран результат функции `max_num` и случается `break` этого `switch`. Если `arr[0]==1` функция печатает на экран результат функции `min_num` и ломается функция. И таким образом если `arr[0]== 2` или `3` печататься на экране `diff_num` или `sum_num` соответственно. В случае `arr[0]` не равен ни какому из `case` тогда выводиться на экран "Данные некорректны".

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 9 2 4 7 6	9	верно

2.	1 9 2 4 7 6	2	верно
3.	2 9 2 4 7 6	7	верно
4.	3 9 2 4 7 6	9	верно

Выводы.

В ходе лабораторной работы были изучены управляющие конструкции в языке Си. Усвоил спопб работы условия, циклы, оператор switch и попытки работы с указателямию

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

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

```
int arr_inp(int arr[]){
    char sep;
    int n = 0;
    for (int i = 0; i < 100; i++){
        scanf("%d%c", &arr[i], &sep);
        n++;
        if(sep == '\n'){
            break;
        }
    }
    return n;
}
```

```
int max_num(int arr[], int n){
    int max = arr[1];
    for(int i=1;i < n;i++){
        if(arr[i]>=max){
```

```

        max = arr[i];
    }
}
return max;
}

```

```

int min_num(int arr[], int n){
    int min = arr[1];
    for(int i=1;i < n;i++){
        if(arr[i]<=min){
            min = arr[i];
        }
    }
    return min;
}

```

```

int diff_num(int arr[], int n){
    int max = max_num(arr, n);
    int min = min_num(arr, n);
    int diff = max - min;
    return diff;
}

```

```

int sum_num(int arr[], int n){
    int sum = 0;
    int min = min_num(arr , n);
    for(int i = 1; i < n; i++){
        if(arr[i] == min){
            break;
        }
    }
}

```

```

        }
        sum = sum + arr[i];

    }
    return sum;
}

int main(){
    int arr[100];
    int n = arr_inp(arr);
    switch(arr[0]) {
        case 0:
            printf("%d\n", max_num(arr, n));
            break;
        case 1:
            printf("%d\n", min_num(arr, n));
            break;
        case 2:
            printf("%d\n", diff_num(arr, n));
            break;
        case 3:
            printf("%d\n", sum_num(arr, n));
            break;
        default:
            printf("Данные некорректны\n");
    }
    return 0;
}

```