

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

Поршнев Р.А.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Исследование управляющих конструкций языка Си, изучение основных способом написания программы и начало изучения функционального программирования.

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".

Вариант 2.

Основные теоретические положения.

В данной лабораторной работе использовалась библиотека *stdio.h*.

Выполнение работы.

В функции *main()* объявляется массив *arr[N]* типа данных *int*, $N = 100$ — это размерность массива. После массива в этой же функции объявляется переменная *n* типа *int*, которой приравниваем значение функции *readarr(int *arr)*.

В функцию *readarr(int *arr)* передаётся массив *arr*. В данной функции объявляется счётчик в цикле *i* и счётчик количества элементов массива *arr*, который мы назовём *n*. Оба счётчика типа *int*. Также объявляем переменную *c* типа *char*, которая нужна для определения конца ввода массива.

```
for(i = 0; i < N; i++) {  
    n = n + 1;  
    scanf("%d%c", &arr[i], &c);
```

```

        if (c == '\n') {
            break;
        }
    }
}

```

То есть, сначала выполняется ввод элемента массива *arr[i]*. Если мы хотим ввести ещё один элемент массива, то следует поставить пробел и выполнить ввод следующего элемента. Если мы больше не собираемся вводить элемент массива, то просто выполняем переход на следующую строку и цикл завершается с помощью оператора *break*. После этого функции *readarr(int *arr)* возвращается значение *n* — количество элементов массива.

После чего в функции *main()* используем оператор *switch* для первого введённого значения массива. В зависимости от этого значения вызывается одна из функций, каждую из которых рассмотрим подробнее.

```

switch(arr[0]) {
    case 0:
        printf("%d\n", max(arr, n));
        break;
    case 1:
        printf("%d\n", min(arr, n));
        break;
    case 2:
        printf("%d\n", diff(arr, n));
        break;
    case 3:
        printf("%d\n", sum(arr, n));
        break;
    default:
        printf("Данные некорректны\n");
}

```

Если введённое значение равно 0, то вызывается функция *max(int *arr, int n)*. В качестве аргументов передаётся массив *arr* и количество элементов массива *n*. В данной функции объявляется счётчик в цикле *i* и переменная *maxinarr*, в которую будем записывать максимальный элемент массива. Перед циклом переменной *maxinarr* присвоим значение *arr[1]*. В цикле будет находиться условие: если элемент массива *arr[i]* больше либо равен максимальному значению элемента массива на момент прошлых итераций *maxinarr*, то *maxinarr* присвоим *arr[i]*. После последней итерации цикла функции *max(int *arr, int n)* вернём значение *maxinarr*.

```

for(i = 1; i < n; i++) {
    if (arr[i] >= maxinarr) {
        maxinarr = arr[i];
    }
}

```

```

    }
    return maxinarr;

```

Если введённое значение равно 1, то вызывается функция *min(int *arr, int n)*. В качестве аргументов передаётся массив *arr* и количество элементов массива *n*. В данной функции объявляется счётчик в цикле *i* и переменная *mininarr*, в которую будем записывать минимальный элемент массива. Перед циклом переменной *mininarr* присвоим значение *arr[1]*. В цикле будет находиться условие: если элемент массива *arr[i]* меньше либо равен минимальному значению элемента массива на момент прошлых итераций *mininarr*, то *mininarr* присвоим *arr[i]*. После последней итерации цикла функции *min(int *arr, int n)* вернём значение *mininarr*.

```

for(i = 1; i < n; i++) {
    if (arr[i] <= mininarr) {
        mininarr = arr[i];
    }
}
return mininarr;

```

Если введённое значение равно 2, то вызывается функция *diff(int *arr, int n)*. В качестве аргументов передаётся массив *arr* и количество элементов массива *n*. В данной функции объявляются переменные *maxinarr*, *mininarr*, *x* типа *int*. Переменной *maxinarr* присваивается значение функции *max(int *arr, int n)*, *mininarr* присваивается значение функции *min(int *arr, int n)*, переменной *x* присваивается разность между *maxinarr* и *mininarr*. Функции *diff(int *arr, int n)* возвращаем *x*.

```

maxinarr = max(arr, n);
mininarr = min(arr, n);
x = maxinarr - mininarr;
return x;

```

Если введённое значение равно 3, то вызывается функция *sum(int *arr, int n)*. В качестве аргументов передаётся массив *arr* и количество элементов массива *n*. В данной функции объявляются переменные *i*, *x*, *summa* типа *int*. Переменная *i* выполняет функцию счётчика в цикле, переменной *x* присваиваем значение функции *min(int *arr, int n)*, а в переменную *summa* будем записывать сумму элементов массива до того момента, пока элемент массива не равен минимальному. Если элемент массива равен минимальному, то цикл сразу же завершается.

```

x = min(arr, n);

```

```

summa = 0;
for(i = 1; i < n; i++) {
    if (arr[i] == x) {
        break;
    }
    summa = summa + arr[i];
}

```

После цикла функции *sum(int *arr, int n)* возвращаем переменную *summa*.

При любом другом вводимом значении выводится: “Данные некорректны”.

Стоит упомянуть: в функциях *min(int *arr, int n)*, *max(int *arr, int n)*, *diff(int *arr, int n)* и *sum(int *arr, int n)* в циклах индексация идёт с единицы, ведь *arr[0]* является значением, исходя из которого будут определены дальнейшие действия с массивом *arr*.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 3 1 4 6 10 7 -1 2 -100	10	Ответ правильный
2.	0 10 4 7 89 100 2 1 4 3	100	Ответ правильный
3.	1 10 89 67 89 999 1000000	10	Ответ правильный
4.	1 13 89 67 89 1 12300 900	1	Ответ правильный
5.	2 3 1 4 2 6 4 8 6 10 11 12 20	19	Ответ правильный
6.	2 100 89 67 65 -100	200	Ответ правильный
7.	3 3 1 4 6 10 7 -1 2 -100	32	Ответ правильный
8.	3 1 2 3 4 5 6 7 8 9 10 11 12	0	Ответ правильный
7.	10 3 1 4 6 10 7 -1 2 -100	Данные некорректны	Ответ правильный
8.	-66 174 477 2 848 184	Данные некорректны	Ответ правильный

Выводы.

Я исследовал управляющие конструкции языка Си, изучил основные способы написания программы и начало функционального программирования.

Были изучены простейшие арифметические операции, условные операторы, циклы, массивы.

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. Далее с помощью оператора switch программа выполняла определённые действия в зависимости от начального значения. Для обработки данных использовались простейшие арифметические операции и условный оператор if.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab1.c

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

```
#define N 100
```

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

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

```
int diff(int *arr, int n) {  
  
    int maxinarr, mininarr, x;  
    maxinarr = max(arr, n);  
    mininarr = min(arr, n);  
    x = maxinarr - mininarr;  
    return x;  
}
```

```
int sum(int *arr, int n) {  
  
    int i, x, summa;  
    x = min(arr, n);  
    summa = 0;  
    for(i = 1; i < n; i++) {  
        if (arr[i] == x) {  
            break;  
        }  
    }  
    summa = summa + arr[i];  
}
```

```

return summa;

}

int readarr(int *arr) {

    int i, n = 0;
    char c;
    for(i = 0; i < N; i++) {
        n = n + 1;
        scanf("%d%c", &arr[i], &c);
        if (c == '\n') {
            break;
        }
    }
    return n;
}

int main() {

    int arr[N];
    int n;
    n = readarr(arr);
    switch(arr[0]) {
        case 0:
            printf("%d\n", max(arr, n));
            break;
        case 1:
            printf("%d\n", min(arr, n));
            break;
        case 2:
            printf("%d\n", diff(arr, n));
            break;
        case 3:
            printf("%d\n", sum(arr, n));
            break;
        default:
            printf("Данные некорректны\n");
    }
    return 0;
}

```