

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. 0382		Шангичев В. А.
Преподаватель		Жангиров Т. Р.

Санкт-Петербург
2020

Цель работы.

Ознакомиться с базовыми синтаксическими конструкциями языка Си.

Задание.

Вариант №2

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".

Основные теоретические положения.

В программе использовались следующие конструкции:

- Стандартная библиотека Си `studio.h`;
- Именованная константа, объявленная с помощью синтаксической конструкции `#define`;
- Функции `max`, `min`, `diff` и `main`;
- Для вывода и ввода данных использовались функции `scanf` и `printf`;
- Также для хранения данных в программе были использованы переменные и целочисленный массив;
- Логические конструкции `switch`, `else`, `if`;
- `break`;
- Циклы `while` и `for`;

Выполнение работы.

1. Функция `max`.

Цель данной функции получать на вход массив и его размер и возвращать максимальное число в массиве. Сначала объявляются переменные `i` и `maximum`. Переменная `i` будет использоваться для обращения к элементам массива по индексу, а `maximum` - для хранения текущего максимального значения. Далее используется цикл `for`, где перебираются все элементы массива и выбирается максимум, который и будет возвращен в конце выполнения данной функции.

2. Функция `min`.

Данная функция действует по такому же алгоритму, как и функция `max` (см. выше), но её задача заключается в нахождении минимального элемента массива. Существенным отличием будет лишь проверка в цикле `for`.

3. Функция `diff`.

Эта функция должна выводить разницу между максимальным и минимальным элементом в массиве. Для нахождения максимума и минимума вызываются описанные выше функции `min` и `max`. После этого вычисляется разность этих значений и возвращается в качестве результата.

4. Функция `sum`.

В данной функции мы считаем сумму элементов массива, расположенных до первого минимального элемента. Сначала мы вычисляем минимум, используя функцию `min`. Значение суммы будет сохраняться в переменной `sum`. При помощи цикла `for` перебираются все элементы массива. На каждой итерации мы проверяем, не равен ли этот элемент найденному минимуму. Если нет, то значение

прибавляется к сумме, и выполнение цикла продолжается. Если да, то выполнение программы прекращается с помощью возвращения значения текущей суммы.

5. Именованная константа.

В коде также присутствует именованная константа `#define MAXLEN`, которая впоследствии будет использована для задания размера статического массива.

6. Подключение стандартной библиотеки.

В начале файла подключается заголовочный файл `stdio.h`, являющийся стандартным заголовочным файлом языка Си.

7. Функция `main`.

Выполнение программы начинается с функции `main`. В начале программы объявляются несколько переменных и целочисленный массив. Затем пользователю предлагается ввести число от 1 до 4 (результат сохраняется в переменную `query`). Далее с помощью цикла `while` вводятся элементы массива. Итерации выполняются до тех пор, пока мы не превысим размер массива или символом разделения окажется не пробел. В каждой итерации в массиве сохраняется целочисленное значение, а переменной `sep` присваивается строка, идущая вслед за этим значением. Следующим шагом объявляется оператор `switch`, в котором будет отслеживаться значение переменной `query`. В каждом логическом блоке `case` вызывается нужная функция, и значение, возвращенное ей, сохраняется в переменной `answer`. После чего данное значение печатается на экран. Если же переменная не равняется ни одному из допустимых значений, то на экран выводится сообщение о том, что данные некорректны.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1	0 6 92 70 59 63 90 38 57 29 0 47 81 5 48 93 29 69 93 56 23 59 25 33 91 77 71 34 85 62 41 84 66 8 41 25 89 13 68	93	С данной задачей программа справилась успешно.

Выводы.

Удалось успешно применить и, тем самым закрепить знания базовых синтаксических конструкций языка Си.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Исходный код программы хранится в .../src/main.c

```
#include <stdio.h>
#define MAX_LEN 100

int max(int arr[], int size){
    int i = 0, maximum;
    maximum = arr[i];

    for (i = 1; i < size; i++){
        if (arr[i] > maximum) {
            maximum = arr[i];
        }
    }
    return maximum;
}

int min(int arr[], int size){
    int i = 0;
    int minimum;
    minimum = arr[i];

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

int diff(int arr[], int size){
    int diff;
    diff = max(arr, size) - min(arr, size);
}
```

```

        return diff;
    }

int sum(int arr[], int size){
    int minimum = min(arr, size);
    int sum = 0;
    int i;
    for (i = 0; i < size; i++){
        if (arr[i] != minimum){
            sum += arr[i];
        }
        else {
            break;
        }
    }
    return sum;
}

int main() {
    int query, size, answer;
    int values[MAX_LEN];
    char sep = ' ';
    size = 0;

    scanf("%d", &query);
    while (size < MAX_LEN && sep == ' '){
        scanf("%d%c", &values[size++], &sep);
    }

    switch (query){
        case 0:
            answer = max(values, size);
            printf("%d\n", answer);
            break;
        case 1:
            answer = min(values, size);

```

```
        printf("%d\n", answer);
        break;
case 2:
    answer = diff(values, size);
    printf("%d\n", answer);
    break;
case 3:
    answer = sum(values, size);
    printf("%d\n", answer);
    break;
default:
    printf("Данные некорректны\n");
    break;
}
return 0;
}
```