

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

Кривоченко Д. И.

Преподаватель

Чайка К. В.

Санкт-Петербург

2021

Цель работы.

Научиться работать с циклами, условиями, оператором switch в языке C.

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 по модулю число в массиве. (abs_max)

1: минимальное по модулю число в массиве. (abs_min)

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

3: сумму элементов массива, расположенных после максимального по модулю элемента (включая этот элемент). (sum)

Иначе необходимо вывести строку "Данные некорректны".

Вариант – 5.

Основные теоретические положения.

Используется стандартный заголовочный файл ввода-вывода `stdio.h`, функции, оператор `switch`, циклы и условия.

Выполнение работы.

Поставлена задача написать программу, выполняющую операции над числами в массиве. Помимо `main()`, программа содержит 4 функции, выполняющие операции над числами в массиве. Главная функция `int main()` берёт на себя ввод данных массива `arr[N]`, подсчитывает количество введенных чисел в переменную `count`. Далее с помощью функций записываются ответы на задачу, поставленную пользователем (в переменной `usr`) в `answ0`, `answ1`, `answ2`, `answ3` соответственно и вызывается оператор `switch` для вывода ответа в консоль.

Функция `abs_max()` берет на вход массив и его размер, после чего перебором находит максимум по модулю и возвращает его.

Функция `abs_min()` берет на вход массив и его размер, после чего перебором находит минимум по модулю и возвращает его.

Функция `diff()` берет на вход массив и его размер, используя функции `abs_max()` и `abs_min()` находит разность между максимальным и минимальным по модулю числами.

Функция `sum()` получает на вход массив и его размер, а также максимальное по модулю число, после чего вычисляет сумму элементов массива начиная с этого числа.

Переменные:

N – макрос числа 101

c – переменная для проверки конца ввода.

$arr[N]$ – массив чисел, получаемых на вход размера N

usr – определение подзадачи, которую нужно выполнить, используется в операторе `switch`

$count$ – количество введенных элементов массива, используется для подачи на вход функциям

$answ0$ – хранит ответы на задания подзадачу 0, используются для вывода ответа на неё

$answ1$ – хранит ответы на задания подзадачу 0, используются для вывода ответа на неё

$answ2$ – хранит ответы на задания подзадачу 0, используются для вывода ответа на неё

$answ3$ – хранит ответы на задания подзадачу 0, используются для вывода ответа на неё

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1	0 1 5 2 3 3 -10 3 8 7 -10 9 8 7 2 4 14 8 7 2 40 1 8 2 -7 2 1 -2 3 4 8 4 12 2	40	Тест прошёл успешно
2	1 1 5 2 3 3 -10 3 8 7 -10 9 8 7 2 4 14 8 7 2 40 1 8 2 -7 2 1 -2 3 4 8 4 12 2	1	Тест прошёл успешно
3	2 1 5 2 3 3 -10 3 8 7 -10 9 8 7 2 4 14 8 7 2 40 1 8 2 -7 2 1 -2 3 4 8 4 12 2	39	Тест прошёл успешно
4	3 1 5 2 3 3 -10 3 8 7 -10 9 8 7 2 4 14 8 7 2 40 1 8 2 -7 2 1 -2 3 4 8 4 12 2	78	Тест прошёл успешно
5	4 1 5 2 3 3 -10 3 8 7 -10 9 8 7 2 4 14 8 7 2 40 1 8 2 -7 2 1 -2 3 4 8 4 12 2	Данные некорректны	Тест прошёл успешно

Выводы.

В результате работы было закреплено знание основных конструкций языка, исследованы условия, циклы, операторы на практике.

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. Первым значением является подзадача, последующими – числа в массиве, с которыми нужно осуществить работу в зависимости от подзадачи. Задача решена с помощью условий if-else для обработки входных данных, циклов for, while для работы с массивом, оператора switch для выбора подзадачи.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

```
#include <stdlib.h>
```

```
#define N 101
```

```
int abs_max(int list[], int k);      //max по модулю
```

```
int abs_min(int list[], int k);      //min по модулю
```

```
int diff(int list[], int k);         //разница между max и min
```

```
int sum(int list[], int k, int start_num); //сумма элементов, начина с max по модулю
```

```
int main()
```

```
{
```

```
    char c;
```

```
    int arr[N];
```

```
    int usr;
```

```
    int count = 0;
```

```
    int answ0, answ1, answ2, answ3;
```

```
    scanf("%d\n", &usr);
```

```
    while (count <= N)
```

```
    {
```

```
        scanf("%d%c", &arr[count], &c);
```

```
        count++;
```

```
        if (c == '\n')
```

```
        {
```

```
            break;
```

```
        }
```

```

    if (count > (N - 1))
    {
        printf("Данные некорректны");
        return 0;
    }
}

```

```

answ0 = abs_max(arr, count);
answ1 = abs_min(arr, count);
answ2 = diff(arr, count);
answ3 = sum(arr, count, answ0);

```

```

switch (usr)
{
case 0:
    printf("%d\n", answ0);
    break;
case 1:
    printf("%d\n", answ1);
    break;
case 2:
    printf("%d\n", answ2);
    break;
case 3:
    printf("%d\n", answ3);
    break;
default:
    puts("Данные некорректны");
}

return 0;

```

```
}
```

```
int abs_max(int list[], int k)
{
    int max_num = 0;
    int answ;
    for (int i = 0; i < k; i++)
    {
        if (abs(list[i]) > max_num)
        {
            max_num = abs(list[i]);
            answ = list[i];
        }
    }
    return answ;
}
```

```
int abs_min(int list[], int k)
{
    int min_num = 10000;
    int answ;
    for (int i = 0; i < k; i++)
    {
        if (abs(list[i]) < min_num)
        {
            min_num = abs(list[i]);
            answ = list[i];
        }
    }
    return answ;
}
```

```

int diff(int list[], int k)
{
    int ma = abs_max(list, k);
    int mi = abs_min(list, k);
    return (ma - mi);
}

int sum(int list[], int k, int start_num)
{
    int tumblr = 0;
    int curr_sum = 0;
    for (int i = 0; i < k; i++)
    {
        if (list[i] == start_num)
        {
            tumblr = 1;
        }
        if (tumblr)
        {
            curr_sum = curr_sum + list[i];
        }
    }
    return curr_sum;
}

```