

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

Спасов Д.В.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Изучение базовых конструкций языка Си. Применение этих конструкций на практике.

Задание.

Реализовать 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.

Цель работы.

Изучение базовых конструкций языка Си. Применение этих конструкций на практике.

Задание.

Реализовать 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 : индекс первого отрицательного элемента. (index_first_negative)
1 : индекс последнего отрицательного элемента. (index_last_negative)
2 : Найти сумму модулей элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент). (sum_between_negative)
3 : Найти сумму модулей элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент). (sum_before_and_after_negative)

Оформить в виде функций. Вариант 5

Основные теоретические положения.

stdio.h-Стандартная библиотека ввода и вывода

stdlib.h-Библиотека, из которой берем операцию abs()

abs(n)-модуль числа n

printf(<строка форматов>, арг 1, арг 2, ..., арг n)-ВЫВОД

scanf(<строка форматов>, арг 1, арг 2, ..., арг n)-ВВОД

if (<условие>)

{тело}

else

{тело}-условный оператор

while (условие)

{тело} -цикл с предусловием

do

{тело}

while (условие) -цикл с постусловием

switch(выражение)

{ case_1: тело1

case_2:тело2

case_n:тело n

default:тело_def} -оператор множественного выбора

Выполнение работы.

Переменные:

1 `abs_max()` - Принимает на вход массив `arr` и количество переменных в массиве `k`. Возвращает максимальное по модулю число в массиве

2 `abs_min()` - Принимает на вход массив `arr` и количество переменных в массиве `k`. Возвращает минимальное по модулю число в массиве

3 `diff()` - Принимает на вход массив `arr` и количество переменных в массиве `k`. Возвращает Разницу между максимальным по модулю и минимальным по модулю элементами в массиве

4 `sum()` - Принимает на вход массив `arr`, количество переменных в массиве `k` и Максимальный по модулю элемент в массиве `mx_abs`. Возвращает сумму элементов массива, расположенных после максимального по модулю элемента (включая этот элемент).

Тестирование.

Здесь результаты тестирования, которые помещаются на одну страницу.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

| № п/п | Входные данные | Выходные данные | Комментарии |
|-------|---|-----------------|---|
| 1. | 0 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 -26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9 | 30 | Макс по модулю элемент в массиве 30 |
| 2. | 1 30 20 -5 | -5 | Мин по модулю элемент в массиве 30 |
| 3. | 2 40 15 16 | 25 | Разница между макс по модулю элементом(40) и мин по модулю элементом(15) ответ 25 |
| 4 | 3 15 16 20 5 | 25 | Сумма всех элементов массива после макс по модулю элемента(включая его) ответ 25 |

Выводы.

1 Изучены основные управляющие конструкции языка Си

2 Изучены функции в языке Си

3 Разработана программа способная обрабатывать массив и находить в

нем:

1 максимальное по модулю число в массиве.

2 минимальное по модулю число в массиве.

3 разницу между максимальным по модулю и минимальным по модулю элементом.

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

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb1.c

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

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

```
int abs_max(int arr[], int k);
int abs_min(int arr[], int k);
int diff(int arr[], int k);
int sum(int arr[], int k, int mx_abs);

int main()
{
    char inp;
    int arr[101];
    int command;
    int count = 0;
    int mx, mn, df, sm;

    scanf("%d\n", &command);
    while (count <= 101)
    {
        scanf("%d%c", &arr[count], &inp);
        count++;
        if (inp == '\n')
        {
            break;
        }
        if (count > 100)
        {
            printf("Данные некорректны");
            return 0;
        }
    }

    mx = abs_max(arr, count);
    mn = abs_min(arr, count);
    df = diff(arr, count);
    sm = sum(arr, count, mx);

    switch (command)
    {
        case 0:
            printf("%d\n", mx);
            break;
        case 1:
            printf("%d\n", mn);
            break;
        case 2:
            printf("%d\n", df);
            break;
        case 3:
            printf("%d\n", sm);
            break;
        default:
            puts("Данные некорректны");
    }
}
```

```

    }

    return 0;
}

int abs_max(int arr[], int k)
{
    int max_abs = 0;
    int max;
    for (int i = 0; i < k; i++)
    {
        if (abs(arr[i]) > max_abs)
        {
            max_abs = abs(arr[i]);
            max = arr[i];
        }
    }
    return max;
}

int abs_min(int arr[], int k)
{
    int min_abs = 10000;
    int min;
    for (int i = 0; i < k; i++)
    {
        if (abs(arr[i]) < min_abs)
        {
            min_abs = abs(arr[i]);
            min = arr[i];
        }
    }
    return min;
}

int diff(int arr[], int k)
{
    int max = abs_max(arr, k);
    int min = abs_min(arr, k);
    return (max - min);
}

int sum(int arr[], int k, int mx_abs)
{
    int ind = 0;
    int sm = 0;
    for (int i = 0; i < k; i++)
    {
        if (arr[i] == mx_abs)
        {
            ind = 1;
        }
        if (ind)
        {
            sm = sm + arr[i];
        }
    }
    return sm;
}

```

}