

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Программирование»
Тема: Условия, циклы, оператор switch

Студент гр. 9383

Девятериков И.С.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2019

ВАРИАНТ 2

Цель работы.

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.

1 : минимальное число в массиве.

2 : разницу между максимальным и минимальным элементом.

3 : сумму элементов массива, расположенных до первого минимального элемента.

иначе необходимо вывести строку "Данные некорректны".

Код программы.

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

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

```
int main()
```

```
{
```

```
    int *m = (int*)malloc(10 * sizeof(int));
```

```
    int i = 0, j = 0;
```

```
    int max = -1, min = 101, sum = 0, smin = 0;
```

```
                                //Заполнение массива числами
```

```
    while(scanf("%d", &m[i++]) == 1)
```

```
    {
```

```
        if (i % 9 == 0){
```

```
            m = realloc(m, (i + 10) * sizeof(int));
```

```
        }
```

```
    }
```

```
    --i;
```

```
    for(j = 1; j < i; ++j){                                //Перебор всех элементов
```

```
        if (m[0] > 3 || m[j] > 100){ //Проверка на ошибку в вводе
```

```

        m[0]=4;
        break;
    }
    if (m[j] > max){
        max = m[j];        //Сохранение макс числа
    }
    if (m[j] < min){
        min = m[j];        //Сохранение мин числа
        smin = sum;
    }
    sum += m[j];            //Сумма всех чисел до текущего
}

switch(m[0]){              //Вывод в терминал ответа
    case 0:
        printf("%d", max);
        break;
    case 1:
        printf("%d", min);
        break;
    case 2:
        printf("%d", max-min);
        break;
    case 3:
        printf("%d", smin);
        break;
    case 4:
        printf("Данные некорректны");
        break;
}
free(m);
return 0;

```

Обработка результатов эксперимента.

Sample Input:

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

Sample Output:

93

Выводы.

Язык программирования С требует аккуратного обращения с памятью. Невнимательность при выделении свободного места под данные может дорого обойтись. Однако это сполна компенсируется высокой степенью производительности.