

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”

Студентка гр. 1384

Логинова А. Ю.

Преподаватель

Жангиров Т. Р.

Санкт-Петербург
2021

Цель работы.

Применить основные конструкции C в разработке, научиться работать с массивами и операторами.

Задание.

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:

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

иначе необходимо вывести строку "Данные некорректны".

Выполнение работы.

На вход программе подается целое значение, которое инициализируется в переменной *task* и будет указывать на дальнейший вывод программы. Далее, через пробел, программа принимает на вход массив целых чисел, который инициализируется в переменной *user_input*, для которой уже был определен максимальный размер 100, исходя из условий. Во время считывания этого массива также определяется переменная *size*, которая указывает на количество переданных значений. Считывание прекращается, как только программа

принимает на вход $\backslash n$, с помощью функции *fgetc()*, считывающей символ из стандартного потока ввода.

В зависимости от значения *task*, в консоль программы выводится одно из 5 значений.

Если *task* = 0, программа вызывает функцию *index_first_negative*, в которую передает массив целых чисел *user_input* и размер, определенных пользователем значений, в переменной *size*. В функции определен цикл, который ищет первое отрицательное значение в массиве *user_input* и возвращает его индекс.

Если *task* = 1, программа вызывает функцию *index_last_negative*, в которую передает массив целых чисел *user_input* и *size*. В функции определен цикл, который ищет последнее отрицательное значение в массиве *user_input* и возвращает его индекс.

Если *task* = 2, программа вызывает функцию *sum_between_negative*, в которую передает массив целых чисел *user_input*, *size*, а также индексы первого и последнего отрицательных значений *user_input*. В функции определен цикл, который суммирует модули значений массива *user_input* между первым и последним отрицательным значением в переменную *sum* и возвращает ее.

Если *task* = 3, программа вызывает функцию *sum_before_and_after_negative*, в которую передает массив целых чисел *user_input*, *size*, индексы первого и последнего отрицательных значений *user_input*. В функции определен цикл, который суммирует модули значений массива *user_input* до первого отрицательного значения и после последнего. Сумма значений лежит в переменной *sum*, которую и возвращает функция.

Если *task* не равен ни одному из вышеперечисленных значений, программа выводит в консоль «Данные некорректны».

Код программы см. в приложении А.

Тестирование.

№ п/п	Входные данные	Выходные данные	Комментарии
1	0 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	3	Программа работает корректно.
2	5 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	Данные некорректны	Программа должна получить запрос от 0 до 3.
3	1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1 1 1 1 1 2 -5 4 3 9 1	95	Программа проигнорирует все значения, которые не поместились в массив и выведет индекс не последнего отрицательного значения, а последнего отрицательного значения в самом массиве.
4	3 1 1 1 1 1 1 1 1 1	-1	Если все значения положительные, функции не смогут работать корректно и в качестве несуществующего индекса/суммы вернут -1.

Выводы.

Были изучены и исследованы основные конструкции и операторы C. Была разработана программа, считывающая информацию пользователя с клавиатуры, для обработки которой использовались условные операторы if/else, switch, были написаны функции для определения отрицательных значений, а также находилась сумма, удовлетворяющая условиям задачи.

ПРИЛОЖЕНИЕ А.

Исходный код программы.

Название файла: main.c

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

int index_first_negative(int arr[], int size){
    int i;
    for(i = 0; i < size; ++i){
        if (arr[i] < 0)
            return i;
    }
    return -1;
}

int index_last_negative(int arr[], int size){
    int i;
    for(i = size - 1; i >= 0; --i){
        if (arr[i] < 0)
            return i;
    }
    return -1;
}

int sum_between_negative(int arr[], int size, int first_index, int
last_index){
    int i, sum = 0;
    if (first_index == -1 || last_index == -1)
        return -1;
    for(i = first_index; i < last_index; ++i)
        sum += abs(arr[i]);
    return sum;
}

int sum_before_and_after_negative(int arr[], int size, int first_index,
int last_index){
    int i, sum = 0;
    if (first_index == -1 || last_index == -1)
        return -1;
    for(i = 0; i < first_index; ++i)
        sum += abs(arr[i]);
    for(i = last_index; i < size; ++i)
        sum += abs(arr[i]);
    return sum;
}

int main(){
    int task, user_input[100] = {0}, size = 0;
    scanf("%d", &task);
    do{
        scanf("%d", &user_input[size]);
        ++size;
    } while (fgetc(stdin) != '\n' && size < 100);
```

```

switch(task){
    case 0:
        printf("%d\n", index_first_negative(user_input, size));
        break;
    case 1:
        printf("%d\n", index_last_negative(user_input, size));
        break;
    case 2:
        printf("%d\n", sum_between_negative(user_input, size,
index_first_negative(user_input, size),
index_last_negative(user_input, size)));
        break;
    case 3:
        printf("%d\n", sum_before_and_after_negative(user_input,
size,
index_first_negative(user_input, size),
index_last_negative(user_input, size)));
        break;
    default:
        printf("Данные некорректны");
        break;
}
return 0;
}

```