

**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. 3342

Иванов Д. М.

Преподаватель

Глазунов С. А.

Санкт-Петербург

2023

### **Цель работы.**

Научиться использовать основные конструкции языка C: условия, циклы, оператор *switch* и другие. Реализовать с их помощью программу, которая принимает на вход данные и выводит необходимый результат в соответствии с заданием.

### **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 : 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*)

иначе необходимо вывести строку "Данные некорректны".

### **Выполнение работы.**

В начале реализована основная функция программы *int main()*, в которой происходит считывание строки с помощью функции *fgets()*, разбиение её по пробелам на части с помощью *strtok()*, и их приведение к целочисленному типу (функция *atoi()*). Далее происходит создание массива из исходных чисел и

выбор выполнения определенной функции в зависимости от введенного номера.

Для условия 0 написана функция *int index\_first\_negative(int arr[], int N)*, принимающая массив и его длину. Она с помощью цикла находит индекс первого отрицательного элемента и возвращает его.

Для условия 1 - *int index\_last\_negative(int arr[], int N)*, принимающая по аналогии массив и его длину. Начиная с конца массива, функция при помощи цикла возвращает последний отрицательный элемент.

Для условия 2 - *int sum\_between\_negative(int arr[], int N)*, принимающая по аналогии массив и его длину. Используя ранее написанные функции, обозначены границы цикла и высчитан итоговый счетчик суммы модулей.

Для условия 3 - *int sum\_before\_and\_after\_negative(int arr[], int N)*, принимающая по аналогии массив и его длину. В начале посчитана сумма модулей всех элементов массива. Так как *sum\_between\_negative + sum\_before\_and\_after\_negative* = вся сумма модулей, то верным выводом будет разница всей суммы и *sum\_between\_negative*.

#### Константы:

- *ARR\_SIZE* – максимальный размер массива
- *SIZE* – максимальный размер введенной строки чисел

#### Переменные:

- *int number\_of\_func* – введенный пользователем номер функции, которая должна быть выполнена
- *int arr[ARR\_SIZE]* – массив, для хранения введенных чисел
- *int len* – размер массива
- *char str* – введенная строка чисел
- *char \*str0* – указатель на адрес первого числа массива, во время прохождения цикла и новых вызовов функции *strtok()* переменной будут присваиваться адреса следующих элементов из этой структуры
- *int i* – счетчик в цикле

- `int j` – счетчик в цикле
- `int N` – аргумент функции, принимающий длину массива
- `int arr[]` – аргумент функции, принимающий массив чисел
- `int sum` – счетчик суммы модулей элементов массива от первого отрицательного элемента(включительно) до последнего отрицательного(не включительно)
- `int sum_all` – счетчик суммы модулей элементов всего массива

### Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ 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	Верный вывод
2.	1 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	20	Верный вывод
3.	2 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	226	Верный вывод
4.	3 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	30	Верный вывод

### Выводы.

Была разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. Входная строка разбивается на числа, которые потом добавляются в массив. В зависимости от этой команды выполняется тот или иной алгоритм и выводится соответствующий результат. Изучены и проработанны такие конструкции языка C, как условия, циклы и оператор *switch*. Для упрощения работы все варианты алгоритмов написаны в различных функциях.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 20000
#define ARR_SIZE 100

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

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

int sum_between_negative(int arr[], int N){
    int sum = 0;
    for (int i = index_first_negative(arr, N); i <
index_last_negative(arr, N); i++){
        sum += abs(arr[i]);
    }
    return sum;
}

int sum_before_and_after_negative(int arr[], int N){
    int sum_all = 0;
    for (int i = 0; i < N; i++){
        sum_all += abs(arr[i]);
    }
    return sum_all - sum_between_negative(arr, N);
}

int main(){
    int number_of_func;
    char str[SIZE];
    int arr[ARR_SIZE];
    fgets(str, SIZE, stdin);

    char *str0 = strtok(str, " ");
    number_of_func = atoi(str0);
    str0 = strtok(NULL, " ");
    int len = 0;

    while (str0 != NULL)
    {
```

```

        arr[len++] = atoi(str0);
        str0 = strtok(NULL, " ");
    }

    switch (number_of_func){
        case 0:
            printf("%d\n", index_first_negative(arr, len));
            break;

        case 1:
            printf("%d\n", index_last_negative(arr, len));
            break;
        case 2:
            printf("%d\n", sum_between_negative(arr, len));
            break;

        case 3:
            printf("%d\n", sum_before_and_after_negative(arr, len));
            break;

        default:
            printf("Данные некорректны");
            break;
    }

    return 0;
}

```

## ПРИЛОЖЕНИЕ Б

### ТЕСТИРОВАНИЕ

Таблица Б.2 - Примеры тестовых случаев

№ п/п	Входные данные	Выходные данные	Комментарии
5.	0 23 45 -6 7 89 0 -78 -2 42 9 102 -2 56 7	2	Верный вывод
6.	1 23 45 -6 7 89 0 -78 -2 42 9 102 -2 56 7	11	Верный вывод
7.	2 23 45 -6 7 89 0 -78 -2 42 9 102 -2 56 7	335	Верный вывод
8.	3 23 45 -6 7 89 0 -78 -2 42 9 102 -2 56 7	133	Верный вывод