

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

**ОТЧЕТ**  
**по лабораторной работе №2 по**  
**дисциплине «Программирование»**  
**ТЕМА: Условия, циклы, оператор switch**

Студент гр. 6304

Юсковец А.В.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

## Contents

Цель работы .....	3
Ход работы .....	3
Вывод.....	9

## Цель работы

Создать проект с make-файлом: Реализуйте функцию-меню, 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.c*)

1 : индекс последнего отрицательного элемента. (*index\_last\_negative.c*)

2 : Найти сумму модулей элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент). (*sum\_between\_negative.c*)

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.c*)

иначе необходимо вывести строку "Данные некорректны".

## Ход работы

- **index\_first\_negative.h**

---

**#pragma once**

**int** index\_first\_negative(**int const** a[], **int** size);

---

В этом и во всех следующих заголовочных файлах будет использована директива препроцессора **#pragma once**, чтобы избежать многократного включения в файлы с кодом при использовании **#include**. Объявлена функция, возвращающая **int**, и принимающая массив и его размер. Функция возвращает индекс первого отрицательного элемента во входном массиве.

- **index\_last\_negative.h**

---

**#pragma once**

**int** index\_first\_negative(**int const** a[], **int** size);

---

Объявлена функция, возвращающая **int**, и принимающая массив и его размер. Функция возвращает индекс последнего отрицательного элемента во входном массиве.

## • sum\_between\_negative.h

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

int sum_between_negative(int const a[], int size);
```

Подключена библиотека `stdlib.h`, т. к. в дальнейшем в соответствующем файле с кодом будет использоваться функция `abs`, возвращающая модуль числа. Объявлена функция, возвращающая `int`, и принимающая массив и его размер. Функция возвращает сумму всех элементов от первого отрицательного элемента до последнего отрицательного.

## • sum\_before\_and\_after\_negative.h

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

int sum_before_and_after_negative(int const a[], int size);
```

Подключена библиотека `stdlib.h`, т. к. в дальнейшем в соответствующем файле с кодом будет использоваться функция `abs()`. Объявлена функция, возвращающая `int`, и принимающая массив и его размер. Функция возвращает сумму всех элементов от первого до первого отрицательного элемента и от последнего отрицательного до последнего.

## • index\_first\_negative.c

```
#include "index_first_negative.h"

int index_first_negative(int const a[], int size) {
    for (int i = 0; i != size; ++i) {
        if (a[i] < 0) { return i; }
    }

    return -1;
}
```

Подключен соответствующий заголовочный файл с объявлением представленной функции. В теле функции выполняется цикл, в котором индекс `i`, проходя по всему массиву останавливается на первом отрицательном элементе и возвращает индекс, на чем заканчивается выполнение функции, в случае, если в массиве нет отрицательных элементов возвращается `-1`.

## • index\_last\_negative.c

```
#include "index_last_negative.h"

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

    return -1;
}
```

Реализация данной функции аналогична реализации предыдущей функции с тем лишь отличием, что в цикле индекс проходит по массиву с конца.

## • sum\_between\_negative.c

```
#include "sum_between_negative.h"
#include "index_first_negative.h"
#include "index_last_negative.h"

int sum_between_negative(int const a[], int size) {
    int sum = 0;

    for (int i = index_first_negative(a, size); i !=
index_last_negative(a, size); ++i) {
        sum += abs(a[i]);
    }

    return sum;
}
```

Подключен соответствующий файлу с кодом заголовочник, и ранее реализованные функции, для нахождения индексов первого и последнего отрицательных элементов. В первую очередь нулем инициализируется переменная `sum`, затем в цикле индекс проходит от первого отрицательного элемента до последнего отрицательного, на каждой итерации добавляя к `sum` модуль `i`-того элемента.

## • `sum_before_and_after_negative.c`

```
#include "sum_before_and_after_negative.h"
#include "index_first_negative.h"
#include "index_last_negative.h"

int sum_before_and_after_negative(int const a[], int size) {
    int sum = 0;

    for (int i = 0; i != index_first_negative(a, size); ++i) {
        sum += abs(a[i]);
    }
    for (int i = index_last_negative(a, size); i != size; ++i) {
        sum += abs(a[i]);
    }

    return sum;
}
```

Подключен соответствующий файлу с кодом заголовочник, и ранее реализованные функции, для нахождения индексов первого и последнего отрицательных элементов. В первую очередь нулем инициализируется переменная `sum`, затем в первом цикле индекс проходит от первого элемента до первого отрицательного, а во втором цикле от последнего отрицательного элемента до последнего, на каждой итерации добавляя к `sum` модуль `i`-того элемента в обоих циклах.

```
#include <stdio.h>
#include "index_first_negative.h"
#include "index_last_negative.h"
#include "sum_between_negative.h"
#include "sum_before_and_after_negative.h"

int main() {
    int a[100];
    int command;
    scanf("%d", &command);

    int size;
    for (size = 0; getchar() != '\n' && size != 100; ++size) {
        scanf("%d", &a[size]);
    }

    if (size == 0) {
        printf("Данные некорректны\n");
        return 0;
    }

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

    return 0;
}
```

---

Подключаются все ранее описанные заголовочные файлы и библиотека `stdio.h`, для использования функций `scanf`, `printf`, `getchar`. Инициализируется массив длины 100, и целое `command`. Затем с входного потока считывается значение `command`, и в цикле до тех пор, пока не будет введен символ переноса строки или не будет зафиксирован выход из выделенной области памяти (`size != 100`) будет производиться чтение и поочередное записывание вводимых чисел в массив. В случае, если после передачи команды не последовало каких-либо входных данных будет выведено сообщение *"Данные некорректны"*.

Далее оператору **switch** передается `command` и в соответствии с целью работы выводится результат определенной функции. В случае `command` отличающейся от 0, 1, 2, 3 будет выведено сообщение *"Данные некорректны"*.

## • Makefile.c

---

```
all: index_first_negative.o index_last_negative.o
sum_between_negative.o sum_before_and_after_negative.o menu.o
    gcc -std=c99 index_first_negative.o index_last_negative.o
sum_between_negative.o sum_before_and_after_negative.o menu.o -o menu

menu.o: menu.c index_first_negative.h index_last_negative.h
sum_between_negative.h sum_before_and_after_negative.h
    gcc -std=c99 -c menu.c

index_first_negative.o: index_first_negative.c index_first_negative.h
    gcc -std=c99 -c index_first_negative.c

index_last_negative.o: index_last_negative.c index_last_negative.h
    gcc -std=c99 -c index_last_negative.c

sum_between_negative.o: sum_between_negative.c sum_between_negative.h
index_first_negative.h index_last_negative.h
    gcc -std=c99 -c sum_between_negative.c

sum_before_and_after_negative.o: sum_before_and_after_negative.c
sum_before_and_after_negative.h index_first_negative.h
index_last_negative.h
    gcc -std=c99 -c sum_before_and_after_negative.c

clean:
    rm -f *.o menu
```

---

Для каждого файла с кодом была написана цель с командой, компилирующей его. В зависимостях всех этих целей были указаны соответствующие файл с кодом и заголовочники, которые включались в этот файл.

Заведена цель **all**, линкующая все необходимые скомпилированные файлы и называющая выходной исполняемый файл *"menu"*.

Цель **clean** для удаления всех объектных файлов и файла *"menu"*.

Во всех командах указан флаг *"-std=c99"* для того, чтобы можно было инициализировать переменные в цикле *"for"*.



## Вывод

Был создан проект с make-файлом, реализующий требуемый функционал с использованием условных выражений, циклов и оператора **switch**. При компилировании, сборке и выполнении исполняемого файла не возникает ошибок и предупреждений.