

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

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

Студент гр. 1304

Дешура Д.В.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

## **Цель работы.**

Вынести каждую функцию первой лабораторной в отдельный файл и написать Makefile для получившегося проекта.

## **Задание.**

Вариант 6.

В текущей директории создайте проект с make-файлом. Главная цель должна приводить к сборке проекта. Файл, который реализует главную функцию, должен называться menu.c; исполняемый файл - menu. Определение каждой функции должно быть расположено в отдельном файле, название файлов указано в скобках около описания каждой функции.

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".

## **Выполнение работы.**

Функции подзадач программы вынесены в одноимённые файлы с расширением .c, для каждого файла-функции написаны заголовочные .h файлы с определением этих самых функций. К файлам index\_first\_negative.c и

`index_last_negative.c` подключены соответствующие заголовочные файлы: `index_first_negative.h` и `index_last_negative.h`, а для файлов `sum_between_negative.c` и `sum_before_and_after_negative.c` подключены не только одноимённый `.h` файл, но и по 2 `.h` файла предыдущих функций, так как они необходимы для работы текущей функции. Для автоматизации сборки проекта создан `Makefile`.

*n*- целое число, = 100, определяет длину массива чисел `Data_arr[]`;

*number\_of\_command*- целое число, хранит номер команды;

*flag\_2*- целое число, = 0, считает количество введённых отрицательных чисел, если таких меньше 2-х, то введённые данные не корректны;

*result\_of\_programm*- целое число, результат программы;

*Data\_arr[]*- массив целых чисел, хранит данные для обработки;

*ch*- символ, во время ввода считывает пробелы между числами, как только принимает знак переноса строки, ввод прерывается;

*i*- целое число, счётчик цикла `for`;

файл *menu.c*

`int main()` - основная функция программы. Считывает ввод: сначала первое число в переменную *number\_of\_command*, затем все оставшиеся числа по одному в массив *Data\_arr[]*, пока не встретится `'\n'`, параллельно заполнению массива, считаем количество отрицательных среди чисел и записываем результат в переменную *flag\_2*. Далее – проверка данных, если *flag\_2* < 2 (количество отрицательных чисел меньше 2), то выводим “Данные некорректны” и завершаем программу. Затем в зависимости от введённого *number\_of\_command*, вызываем функции и присваиваем их значения переменной *result\_of\_programm*. Действия в зависимости от значения переменной *number\_of\_command*: `index_first_negative()` для 0, `index_last_negative()` для 1, `sum_between_negative()` для 2, `sum_before_and_after_negative()` для 3, для любого другого значения выводим “Данные некорректны” и завершаем программу.

вызывает функции `index_first_negative()`, `index_last_negative()`, `sum_between_negative()`, `sum_before_and_after_negative()`; выводит результат программы.

файл *index\_first\_negative.c*

`int index_first_negative(int Data_arr[], int n)` - функция, вызываемая при `number_of_command == 0`, принимает массив целых чисел `Data_arr` и его размер `n`. Запускает цикл `for(int i = 0; i < n; i++)` перебирает элементы массива, встретив первый отрицательный элемент, возвращает его индекс.

файл *index\_last\_negative.c*

`int index_last_negative(int Data_arr[], int n)` - функция, вызываемая при `number_of_command == 1`, принимает массив целых чисел `Data_arr` и его размер `n`. Вводит целочисленную переменную `result`. Запускает цикл `for(int i = 0; i < n; i++)` перебирает элементы массива, встретив отрицательный элемент, передаёт его индекс переменной `result`, таким образом после прохождения цикла `result` будет хранить индекс последнего такого числа. Возвращает переменную `result`.

файл *sum\_between\_negative.c*

`int sum_between_negative(int Data_arr[], int index_first_negative, int index_last_negative)` - функция, вызываемая при `number_of_command == 2`, принимает массив целых чисел `Data_arr`, функцию `index_first_negative(Data_arr, n)` и функцию `index_last_negative(Data_arr, n)`. Вводит целочисленную переменную `sum = 0`, запускает цикл `for(int index_arr = index_first_negative; index_arr < index_last_negative; index_arr++)`. Цикл проходит от первого отрицательного (включительно), до последнего отрицательного (не включительно), увеличивая `sum` на модуль каждого элемента. После прохода цикла функция возвращает `sum`.

файл *sum\_before\_and\_after\_negative.c*

`int sum_before_and_after_negative(int Data_arr[], int index_first_negative, int index_last_negative, int n)` - функция, вызываемая при `number_of_command == 3`, принимает массив целых чисел `Data_arr`, его размер `n`, функцию `index_first_negative(Data_arr, n)` и функцию `index_last_negative(Data_arr, n)`.

Вводим целочисленную переменную *sum*. Функция состоит из двух циклов. Первый - *for(int index\_arr = 0; index\_arr < index\_first\_negative; index\_arr++)* проходит по массиву данных от начала до первого отрицательного элемента (не включительно), прибавляя к переменной *sum* модуль каждого элемента. Второй цикл *for(int index\_arr = index\_last\_negative; index\_arr < n; index\_arr++)* выполняет те же действия, но двигается от последнего отрицательного элемента (включительно) к концу массива. Функция возвращает значение переменной *sum*.

Разработанный программный код см. в приложении А.

## Тестирование.

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	Пройден
2.	0 12 23 15 4 -20 0 -5 -9 49 -58 5 65 56 48 23 -9 56 -56 55 6 -9 1	4	Пройден
3.	1 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	20	Пройден
4.	1 16 -1 -1 -3 -45 -46 -64 -56 -26 -9 - 3 -56 -5 -7 -74 -25 -753 -5	17	Пройден
5.	2 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	226	Пройден
6.	2 1 23 54 -31 -432 -54 -543 -54 -65 -5 -7 -2 -54 -12 -45 -6 23 34	1304	Пройден
7.	3 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	30	Пройден
8.	3 12 12 35 4 5 4 54 61 8461 5 421 85 4 -1312 4 -234	9397	Пройден
9.	4 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	Данные некорректны	Пройден
10.	2 1 16 2 18 -22 15 3 13 0 6 1 9 24 1 18 15 28 20 17 16 11	Данные некорректны	Пройден

## Выводы.

Выполнив лабораторную работу №2, мы модифицировали проект из лабораторной работы №1, вынеся каждую подзадачу в отдельный файл и написав Makefile для получившегося проекта.

Были изучены общие принципы написания Makefile для сборки целого проекта, созданы заголовочные файлы для каждого подключаемого файла с функциями.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: menu.c

```
#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 index_arr, number_of_command, result_of_programm, flag_2 = 0,
c = 0, n = 100;
    char ch;
    scanf("%d%c", &number_of_command, &ch);
    int Data_arr[n];

    for(index_arr = 0; index_arr < n; index_arr++){
        if(ch != '\n'){
            scanf("%d%c", &Data_arr[index_arr], &ch);
            if(Data_arr[index_arr] < 0){
                flag_2++;
            }
        } else {
            Data_arr[index_arr] = 0;
        }
    }

    if(flag_2 < 2){
        printf("Д а н н ы е   н е к о р р е к т н ы\n");
        return 0;
    }

    switch(number_of_command){
        case 0:
            result_of_programm = index_first_negative(Data_arr, n);
            break;
        case 1:
            result_of_programm = index_last_negative(Data_arr, n);
            break;
        case 2:
            result_of_programm = sum_between_negative(Data_arr,
index_first_negative(Data_arr, n), index_last_negative(Data_arr, n));
            break;
        case 3:
            result_of_programm =
sum_before_and_after_negative(Data_arr, index_first_negative(Data_arr, n),
index_last_negative(Data_arr, n), n);
            break;
        default:
            printf("Д а н н ы е   н е к о р р е к т н ы\n");
            return 0;
    }

    printf("%d\n", result_of_programm);
}
```

```

    return 0;
}

```

### Название файла: index\_first\_negative.c

```

#include "index_first_negative.h"

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

```

### Название файла: index\_first\_negative.h

```

int index_first_negative(int Data_arr[], int n);

```

### Название файла: index\_last\_negative.c

```

#include "index_last_negative.h"

int index_last_negative(int Data_arr[], int n){
    int i, result;
    for(i = 0; i < n; i++){
        if (Data_arr[i] < 0){
            result = i;
        }
    }
    return result;
}

```

### Название файла: index\_last\_negative.h

```

int index_last_negative(int Data_arr[], int n);

```

### Название файла: sum\_between\_negative.c

```

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

int sum_between_negative(int Data_arr[], int index_first_negative,
int index_last_negative){
    int index_arr, sum = 0;
    for(index_arr = index_first_negative; index_arr <
index_last_negative; index_arr++){
        if(Data_arr[index_arr] < 0){
            sum = sum - Data_arr[index_arr];
        }else{
            sum = sum + Data_arr[index_arr];
        }
    }

    return sum;
}

```



### Название файла: sum\_between\_negative.h

```
#include "index_last_negative.h"
#include "index_first_negative.h"
int sum_between_negative(int Data_arr[], int index_first_negative,
int index_last_negative);
```

### Название файла: sum\_before\_and\_after\_negative.c

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

int sum_before_and_after_negative(int Data_arr[], int
index_first_negative, int index_last_negative, int n){
    int index_arr, sum = 0;
    for(index_arr = 0; index_arr < index_first_negative;
index_arr++){
        if(Data_arr[index_arr] < 0){
            sum = sum - Data_arr[index_arr];
        }else{
            sum = sum + Data_arr[index_arr];
        }
    }
    for(index_arr = index_last_negative; index_arr < n; index_arr++){
        if(Data_arr[index_arr] < 0){
            sum = sum - Data_arr[index_arr];
        }else{
            sum = sum + Data_arr[index_arr];
        }
    }
    return sum;
}
```

### Название файла: sum\_before\_and\_after\_negative.h

```
#include "index_first_negative.h"
#include "index_last_negative.h"
int sum_before_and_after_negative(int Data_arr[], int
index_first_negative, int index_last_negative, int n);
```

### Название файла: Makefile

```
menu.o: menu.c index_first_negative.h index_last_negative.h
sum_between_negative.h sum_before_and_after_negative.h
gcc -c menu.c
```

```
index_first_negative.o: index_first_negative.c
index_first_negative.h
gcc -c index_first_negative.c
```

```
index_last_negative.o: index_last_negative.c index_last_negative.h
gcc -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 -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 -c sum_before_and_after_negative.c

clean:
    rm -rf menu.o index_first_negative.o index_last_negative.o
sum_between_negative.o sum_before_and_after_negative.o menu
```