

**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: Сборка программ в Си**

Студент гр. 1304

Новицкий  
М.Д

Преподаватель

Чайка К.В.

Санкт-Петербург  
2021

**Цель работы.**

Научиться использовать Makefile при создании программы

### **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".

*Ошибкой в данном задании считается дублирование кода!*

*Подсказка: функция нахождения модуля числа находится в заголовочном файле `stdlib.h` стандартной библиотеки языка Си.*

*При выводе результата, не забудьте символ переноса строки*

### **Основные теоретические положения.**

Условия

Циклы

Оператор switch

Makefile

### **Выполнение работы.**

В ходе решения задачи, в функции main, в файле menu.c, были определены переменные для входных данных, осуществлен ввод с клавиатуры для определения значения переменной, отвечающей за режим работы программы, а также для определения значений массива. После получения входных данных, было создано условие, которое проверяет значение переменной, которая определяет режим работы программы. В зависимости от значения введенной переменной, идет обработка определённая обработка массива. Для обработки массива было решено сделать четыре отдельных функции, в четырёх отдельных файлах, каждый из которых также имеет заголовочный файл, решающие поставленные задачи, и одну для ввода данных массива. После обработки массива одной из функций, происходит вывод результата на экран, и программа завершается.

Для того чтобы программа работала корректно, с помощью директив #ifndef-#define-#endif в заголовочном файле была создана защита от повторного включения. Далее был создан Makefile для сборки программы.

Переменные: *comand* — определение режима работы программы, *LIM* — максимальный размер массива, *arr[]* - массив целочисленных значений, *ch* — переменная для отслеживания окончания ввода строки, проверяется на наличие в себе символа перевода строки, *N* — размер массива, *i* — индекс элемента массива, *first\_i* — хранит значение индекса первого отрицательного элемента в массиве, *last\_i* — хранит значение индекса последнего отрицательного элемента в массиве, *sum* — хранит сумму модулей чисел массива, либо между первым и последним отрицательным элементом массива, либо до первого отрицательного и после последнего.

Все функции, каждая из которых находится в своём файле а так же имеет свой заголовочный файл, в качестве аргументов принимают *arg[]* - массив

элементов, и N — размер массива (кол-во значащих элементов массива),  
 index\_first\_negative — находит индекс первого отрицательного элемента.  
 index\_last\_negative — находит индекс последнего отрицательного элемента.  
 sum\_between\_negative — находит сумму модулей элементов массива, расположенных от первого отрицательного элемента(включая) и до последнего отрицательного(не включая).  
 sum\_before\_and\_after\_negative — находит сумму модулей элементов массива, расположенных до первого отрицательного элемента(не включая) и после последнего(включая).  
 print\_arr – функция для нахождения N и ввода массива

Разработанный программный код см. в приложении А.

### **Тестирование.**

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 1 2 -3 4 5 -6 7	2	Успешный тест
2.	0 -1 2 3 -4 5	0	Успешный тест
3.	1 -2 3 4 -5 6	3	Успешный тест
4.	2 3 4 -5 6 -7 8 5	11	Успешный тест

5.	3 -1 -4 -5 -6 -7 -8 8 6 5 4 5 -7	-7	Успешный тест
----	-------------------------------------	----	---------------

### **Выводы.**

Были исследованы способы создания программы с помощью Makefile

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. После считывания первого значения, происходит считывание массива, который обрабатывается функцией в зависимости от номера задания, так 0 — нахождение индекса первого отрицательного элемента массива, 1 — нахождение индекса последнего отрицательного массива, 2 — нахождение суммы модулей элементов массива, расположенных от первого отрицательного элемента(включая) и до последнего(невключая), 3 — нахождение суммы модулей элементов массива, расположенных до первого отрицательного элемента(не включая) и после последнего(включая).

Задача была решена при помощи условий if else, применяемых для обработки входных данных, введенных пользователем, цикла for, который был применён в ходе работы с элементами массива, а также оператора switch, определяющего режим работы программы.

## **ПРИЛОЖЕНИЕ А**

### **ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: menu.c

```
#include <stdio.h>
#include "index_first_negative.h"
#include "index_last_neganive.h"
#include "sum_between_negative.h"
#include "sum_befor_and_after_negative.h"
#define LIM 100
```

```
int print_arr(int* arr){
```

```

int i; char ch;
for (i = 0, ch = ' '; ch != '\n'; i++)
    scanf("%d%c", &arr[i], &ch);
return i;
}
int main(){
    int comand;
    scanf("%d",&comand);
    int arr[LIM];
    int N = print_arr(arr);
    switch(comand){
        case 0:
            if(index_first_negative(arr, N) < N){
                printf("%d\n", index_first_negative(arr, N));
            }
            else{
                printf("Данные некорректны\n");
            }
            break;
        case 1:
            if(index_last_negative(arr,N) >= 0){
                printf("%d\n",index_last_negative(arr,N));
            }
            else{
                printf("Данные некорректны\n");
            }
            break;
        case 2:
            printf("%d\n", sum_between_negative(arr,N));
            break;
        case 3:
            printf("%d\n", sum_before_and_after_negative(arr,N));
            break;
        default:
            printf("Данные некорректны\n");
    }
    return 0;
}

```

Название файла: index\_first\_negative.h

```

#ifndef __index_first_negative__
#define __index_first_negative__
int index_first_negative(int*, int);
#endif

```

Название файла: index\_first\_negative.c

```

#include "index_first_negative.h"
int index_first_negative(int* arr,int N){
    int first_i = 0;
    int i;
    for(i = 0; i<N; i++){
        if(arr[i] < 0){
            first_i = i;
            break;
        }
    }
}

```

```

    }
}
return(first_i);
}

```

Название файла: index\_last\_negative.h

```

#ifndef __index_last_negative__
#define __index_last_negative__
int index_last_negative(int*, int);
#endif

```

Название файла: index\_last\_negative.c

#include "index\_last\_negative.h"

```

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

```

Название файла: sum\_between\_negative.h

```

#ifndef __sum_between_negative__
#define __sum_between_negative__
int sum_between_negative(int*, int);
#endif

```

Название файла: sum\_between\_negative.c

```

#include <stdlib.h>
#include "index_first_negative.h"
#include "index_last_negative.h"
int sum_between_negative(int* arr,int N){
    int sum = 0;
    int i;
    for(i = index_first_negative(arr,N); i < index_last_negative(arr,N);
i++){
        sum += abs(arr[i]);
    }
    return(sum);
}

```

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

```

#ifndef __sum_before_and_after_negative__
#define __sum_before_and_after_negative__
int sum_before_and_after_negative(int*, int);
#endif

```

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

```

#include <stdlib.h>
#include "index_first_negative.h"
#include "index_last_negative.h"
#include "sum_before_and_after_negative.h"

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

```

Название файла: Makefile

all: menu

```

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

```

```

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

```