

**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 языке Си**

Студент гр. 1304

Кривоченко Д.И.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

## Цель работы.

Научиться создавать и собирать программу, состоящую из нескольких файлов. Научиться работать с makefile.

## Задание.

### Вариант 5.

В текущей директории создайте проект с *make*-файлом. Главная цель должна приводить к сборке проекта. Файл, который **реализует главную функцию**, должен называться *main.c*; **исполняемый файл** - *main*. 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 : максимальное по модулю число в массиве. (*abs\_max.c*)

1 : минимальное по модулю число в массиве. (*abs\_min.c*)

2 : разницу между максимальным по модулю и минимальным по модулю элементом. (*diff.c*)

3 : сумму элементов массива, расположенных после максимального по модулю элемента (включая этот элемент). (*sum.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.

Makefile – список инструкций для утилиты *make*, которая позволяет собирать проект сразу целиком. Состоит из списка целей, зависимостей и команд:

`#include` – вызывает заголовочный файл в текущий файл исходного кода  
`#define` – позволяет определить макросы или макроопределения.

### Выполнение работы.

В ходе выполнения задания были использованы функции и исходный код из первой лабораторной работы. При этом, каждая функция, описанная в задании, была вынесена в отдельный файл. Также для каждой функции были написаны заголовочные файлы `.h` с соответствующими названиями. Функция `main` записана в файл `main.c`.

В файле `Makefile` последовательно записываем правила и цели, от которых они зависят. При этом учитываем порядок, в котором должны быть скомпилированы программы.

Разработанный программный код см. в приложении А.

### Тестирование.

№ п/п	Входные данные	Выходные данные	Комментарии
1	0 1 5 2 3 3 -10 3 8 7 -10 9 8 7 2 4 14 8 7 2 40 1 8 2 -7 2 1 -2 3 4 8 4 12 2	40	Тест прошёл успешно
2	1 1 5 2 3 3 -10 3 8 7 -10 9 8 7 2 4 14 8 7 2 40 1 8 2 -7 2 1 -2 3 4 8 4 12 2	1	Тест прошёл успешно
3	2 1 5 2 3 3 -10 3 8 7 -10 9 8 7 2 4 14 8 7 2 40 1 8 2 -7 2 1 -2 3 4 8 4 12 2	39	Тест прошёл успешно
4	3 1 5 2 3 3 -10 3 8 7 -10 9 8 7 2 4 14 8 7 2 40 1 8 2 -7 2 1 -2 3 4 8 4 12 2	78	Тест прошёл успешно
5	4 1 5 2 3 3 -10 3 8 7 -10 9 8 7 2 4 14 8 7 2 40 1 8 2 -7 2 1 -2 3 4 8 4 12 2	Данные некорректны	Тест прошёл успешно

### Выводы.

Мы научились собирать и компилировать программу, состоящую из нескольких файлов с помощью утилиты `make`. Освоили работу с заголовочными файлами и собрали `Makefile` для нашей программы.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

*Название файла: menu.c*

```
#include <stdio.h>
#include <stdlib.h>
#include "abs_max.h"
#include "abs_min.h"
#include "diff.h"
#include "sum.h"
#define N 101
```

*Название файла: menu.c*

```
#include <stdio.h>
#include <stdlib.h>
#include "abs_max.h"
#include "abs_min.h"
#include "diff.h"
#include "sum.h"
#define N 101
```

```
int main()
{
    char c;
    int arr[N];
    int usr;
    int count = 0;
    int answ0, answ1, answ2, answ3;

    scanf("%d\n", &usr);
    while (count <= N)
    {
        scanf("%d%c", &arr[count], &c);
        count++;
        if (c == '\n')
        {
            break;
        }
        if (count > (N - 1))
        {
            printf("Данные некорректны");
        }
    }
}
```

```

        return 0;
    }
}

answ0 = abs_max(arr, count);
answ1 = abs_min(arr, count);
answ2 = diff(arr, count);
answ3 = sum(arr, count, answ0);

switch (usr)
{
case 0:
    printf("%d\n", answ0);
    break;
case 1:
    printf("%d\n", answ1);
    break;
case 2:
    printf("%d\n", answ2);
    break;
case 3:
    printf("%d\n", answ3);
    break;
default:
    puts("Данные некорректны");
}

return 0;
}

```

Название файла: *abs\_max.c*

```

#include "abs_max.h"
int abs_max(int list[], int k)
{
    int max_num = 0;
    int answ;
    for (int i = 0; i < k; i++)
    {
        if (abs(list[i]) > max_num)
        {
            max_num = abs(list[i]);
            answ = list[i];
        }
    }
}

```

```
    return answ;  
}
```

Название файла: *abs\_min.c*

```
#include "abs_min.h"  
int abs_min(int list[], int k)  
{  
    int min_num = 10000;  
    int answ;  
    for (int i = 0; i < k; i++)  
    {  
        if (abs(list[i]) < min_num)  
        {  
            min_num = abs(list[i]);  
            answ = list[i];  
        }  
    }  
    return answ;  
}
```

Название файла: *diff.c*

```
#include "diff.h"  
#include "abs_max.h"  
#include "abs_min.h"  
int diff(int list[], int k)  
{  
    int ma = abs_max(list, k);  
    int mi = abs_min(list, k);  
    return (ma - mi);  
}
```

Название файла: *sum.c*

```
#include "sum.h"  
int sum(int list[], int k, int start_num)  
{  
    int tumbler = 0;  
    int curr_sum = 0;  
    for (int i = 0; i < k; i++)  
    {  
        if (list[i] == start_num)  
        {  
            tumbler = 1;  
        }  
        if (tumbler)  
        {  
            curr_sum = curr_sum + list[i];  
        }  
    }  
}
```

```

    }
}
return curr_sum;
}

```

Название файла: *abs\_max.h*

```

#include <stdio.h>
#include <stdlib.h>
int abs_max(int list[], int k);

```

Название файла: *abs\_min.h*

```

#include "sum.h"
int sum(int list[], int k, int start_num)
{
    int tumblr = 0;
    int curr_sum = 0;
    for (int i = 0; i < k; i++)
    {
        if (list[i] == start_num)
        {
            tumblr = 1;
        }
        if (tumblr)
        {
            curr_sum = curr_sum + list[i];
        }
    }
    return curr_sum;
}

```

Название файла: *diff.h*

```

#include <stdio.h>
#include <stdlib.h>
int diff(int list[], int k);

```

Название файла: *sum.h*

```

#include <stdio.h>
#include <stdlib.h>
int sum(int list[], int k, int start_num);

```