

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(ка) гр. 1304

Спасов Д.В.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Изучение способов сборки программ, а также изучение прекомпиляции компиляции и линковки. Знакомство с утилитой make, заголовочными и объектными файлами.

Задание.

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 : 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!

Основные теоретические положения.

1) Заголовочные файлы в языке C задаются расширением `.h`

2)Объектные файлы являются результатом работы компиляции и задаются расширением .o

3)Makefile служит файлом,в котором просто собирать программу,состоит из цели,зависимости и исполнения:

цель: зависимости

[tab] команда

Выполнение работы.

Программа состоит из 5 файлов, в каждом файле храниться функция отвечающая за конкретную задачу

Функция *abs_max*(строка ввода данных) Получает в качестве входных данных массив состоящий из значений заданных пользователем и переменную, хранящую информацию о количестве элементов в массиве

Результатом работы функции является максимальное по модулю число в массиве

Переменные функции *abs_max*:

max — Предназначена для хранения максимального по модулю числа в массиве

arr — Массив предназначенный для хранения данных, введенных пользователем

k — Предназначена для хранения информации и о количестве элементов массива arr

i — Переменная цикла

Функция ***abs_min***(строка ввода данных) Получает в качестве входных данных массив состоящий из значений заданных пользователем и переменную, хранящую информацию о количестве элементов в массиве

Результатом работы функции является минимальное по модулю число в массиве

Переменные функции *abs_min*:

arr — Массив предназначенный для хранения данных, введенных пользователем

k — Предназначена для хранения информации и о количестве элементов массива arr

i — Переменная цикла

min — предназначена для хранения минимального по модулю числа в массиве

Функция ***diff***(строка ввода данных) Получает в качестве входных данных массив состоящий из значений заданных пользователем и переменную, хранящую информацию о количестве элементов в массиве

Результатом работы функции является разница между максимальным по модулю и минимальным по модулю элементом.

Переменные функции *diff*:

arr — Массив предназначенный для хранения данных, введенных пользователем

k — Предназначена для хранения информации и о количестве элементов массива arr

i — Переменная цикла

mx — Предназначена для хранения данных о максимальное по модулю числе в массиве

mn - предназначена для хранения минимального по модулю числа в массиве

$diff$ — Предназначена для хранения разницы между максимальным по модулю и минимальным по модулю элементами.

Функция *sum*(строка ввода данных) Получает в качестве входных данных массив состоящий из значений заданных пользователем и переменную, хранящую информацию о количестве элементов в массиве

Результатом работы функции является сумма элементов массива, расположенных после максимального по модулю элемента

Переменные функции *sum*:

arr — Массив предназначенный для хранения данных, введенных пользователем

k — Предназначена для хранения информации и о количестве элементов массива arr

i — Переменная цикла

su — Предназначена для хранения суммы элементов массива, расположенных после максимального по модулю элемента

$number$ — Предназначена для хранения данных о номере элемента начиная с которого ведется подсчет

mx — Переменная предназначенная для хранения данных о максимальном по модулю элементе массива

Функция *main()* - является основной функцией программы. Отвечает за считывание данных с клавиатуры, определения количества введенных данных и вызов необходимой функции с последующим выводом результата

Переменные функции *main*:

mn — Предназначена для хранения минимального по модулю числа в массиве

mx — Предназначена для хранения данных о максимальном по модулю элементе массива

su — Предназначена для хранения суммы элементов массива, расположенных после максимального по модулю элемента

dif — Предназначена для хранения разницы между максимальным по модулю и минимальным по модулю элементами.

operation — Предназначена для хранения данных о том, какую именно функцию необходимо вызвать

value — Предназначена для проверки файла на окончание

arr — Массив предназначенный для хранения данных, введенных пользователем

k — Предназначена для хранения информации и о количестве элементов массива *arr*

Тестирование.

Здесь результаты тестирования, которые помещаются на одну страницу.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1	0 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 -	30	Вызывается функция <i>abs_max</i> , которая

	26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9		возвращает максимальное по модулю значение
2	2 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 - 26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9	32	Вызывается функция diff, которая возвращает разность между максимальным по модулю значением и минимальным по модулю значением.
3	3 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 - 26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9	-124	Вызывается функция sum, которая возвращает сумму значений начиная с максимального по модулю значения.
4	1 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 - 26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9	-2	Вызывается функция abs_min, которая возвращает минимальное по модулю значение.

Выводы.

Первоначальное ознакомление с основами сборки программы, а также знакомство с утилитой make и успешная реализация разбиения функций на отдельные самостоятельные программы. Изучены новые расширения .o .h.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Сначала указываем имя файла, в котором код лежит в репозитории:

Название файла:abs_max.c

```
#include <stdio.h>
```

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

```
#include "abs_max.h"
```

```
int abs_max(int arr[], int k){  
    int max = 0;  
    int i;  
    for(i = 0; i < k; i++){  
        if(abs(max) < abs(arr[i])){  
            max = arr[i];  
        }  
    }  
    return max;  
}
```

Название файла:int abs_max.h

```
#include <stdio.h>
```

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

```
int abs_max(int arr[], int k);
```

Название файла:Makefile

```
all: menu.o diff.o sum.o abs_max.o abs_min.o
```

```
gcc menu.o diff.o sum.o abs_max.o abs_min.o -o menu
```

```
menu.o: menu.c abs_max.h abs_min.h diff.h sum.h
```



```
gcc -c menu.c
abs_max.o: abs_max.c abs_max.h
gcc -c abs_max.c
abs_min.o: abs_min.c abs_min.h
gcc -c abs_min.c
diff.o: diff.c diff.h abs_max.h abs_min.h
gcc -c diff.c
sum.o: sum.c sum.h
gcc -c sum.c
clean:
rm *o. a.out
```

Название файла: menu.c

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

```
int main(){
    int k = 0, arr[100], operation;
    int mx, mn, su, dif;
    char value;
    scanf("%d", &operation);

    while(k < 100){
        scanf("%d%c", &arr[k], &value);
        k++;
        if (value == '\n'){
```

```

        break;
    }
}

mn = abs_min(arr, k);
mx = abs_max(arr, k);
su = sum(arr, k);
dif = diff(arr, k);

switch(operation){
    case 0:
        printf("%d\n", mx);
        break;

    case 1:
        printf("%d\n", mn);
        break;

    case 2:
        printf("%d\n", dif);
        break;

    case 3:
        printf("%d\n", su);
        break;

    default: puts("Данные некорректны\n");
}
return 0;
}

```

Название файла:abs_min.c

```
#include <stdio.h>
```

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

```
#include "abs_min.h"
```

```
int abs_min(int arr[], int k){  
    int min = arr[0];  
    int i;  
    for(i = 0; i < k; i++){  
        if(abs(min) > abs(arr[i])){  
            min = arr[i];  
        }  
    }  
    return min;  
}
```

Название файла:abs_min.h

```
#include <stdio.h>
```

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

```
int abs_min(int arr[], int k);
```

Название файла:diff.c

```
#include <stdio.h>
```

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

```
#include "abs_max.h"
```

```
#include "abs_min.h"
```

```
#include "diff.h"
```

```
int diff(int arr[], int k){
```

```
    int diff;
```

```

    int mx = abs_max(arr, k);
    int mn = abs_min(arr, k);
    diff = mx - mn;
    return diff;
}

```

Название файла:diff.h

```

#include <stdio.h>
#include <stdlib.h>

```

```

int diff(int arr[], int k);

```

Название файла:sum.c

```

#include <stdio.h>
#include <stdlib.h>
#include "sum.h"

```

```

int sum(int arr[], int k){
    int su = 0, mx = 0, number;
    int i;
    for(i = 0; i < k; i++){
        if(abs(mx) < abs(arr[i])){
            mx = arr[i];
            number = i;
        }
    }
    for(i = number; i < k; i++){
        su = su + arr[i];
    }
    return su;
}

```

Название файла:sum.h

```
#include <stdio.h>
```

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

```
int sum(int arr[], int k);
```