

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

Нго Тхи Йен

Преподаватель

Чайка К. В.

Санкт-Петербург

2021

Цель работы.

Вынести подзадачи 1 лабораторной работы в отдельные файлы и написать make-файл для получившегося проекта.

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 `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 число в массиве. (`abs_max.c`)

1 : минимальное число в массиве. (`abs_min.c`)

2 : разницу между максимальным и минимальным элементом. (`diff.c`)

3 : сумму элементов массива, расположенных до минимального элемента. (`sum.c`)

ИНАЧЕ : необходимо вывести строку "Данные некорректны".

Выполнение работы.

Для выполнения данной лабораторной работы мы будем использовать код, который был написан ранее для Лабораторной Работы 1 и поэтому, дабы сэкономить время, я не буду углублять в то, как именно работает та или иная функция. На этот раз нам предстоит разбить наши использованные ранее функции на отдельные файлы. Для этого мы сначала создаем файлы при помощи команды `touch` в терминале Linux (*все действия производятся по пути нашей папке(`Ngo_Thi_lb2`), в которых у нас будет храниться определение каждой

функции (touch abs_min.c; touch abs_max.c; touch diff.c; touch sum.c), потом создаем основной файл, который будет их всех задеять(touch menu.c). После этого мы по аналогии создаем файлы, которые будут служить нам для объявления каждой из функций (touch min.h; touch max.h; touch diff.h; touch sum.h). Вскоре мы уже создаем при помощи той же команды touch, файл под название Makefile, в котором у нас будут храниться инструкции по сборке. Далее мы, естественно, убираем из нашей основной программы(main.c) все функции, которые я пометил как *Option Files и добавляем дополнительные директивы с названием наших файлов, объявляющих функции(#include "min.h"; #include "max.h"; #include "diff.h"; #include "sum.h").

Теперь приступим к написанию наших функций. Для этого мы так же заходим в файлы при помощи команды текстового редактора (vi abs_min.c ~ nano abs_min.c), куда мы пока что вставим код с нашими функциями. Далее мы для каждой функции пишем директивы с прототипом функций, соответствующие названиям этих самых функций(т.е. для файла min.c мы пишем #include "min.h"). Важно отметить, что для файлов diff.c и sum.c мы так же пишем и другие директивы(для diff.c мы дополнительно пишем #include "min.h" и #include "max", а для sum.c - #include "min.h"), потому что в коде этих функций мы так же используем первые две функции. Далее приступим к написанию файлов, объявляющих каждую из функций (Единственное их различие будет лишь последняя строка). Для этого мы сначала объявляем в каждом файле директиву stdio(#include <stdio.h>), далее вводим #define N 100 и наконец пишем int <название файла с описанием функции>();

ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ.

СОЗДАНИЕ ПРОЕКТА

1. Создание max.h

```
1
2 int abs_max(int arr[], int size);
```

2. Создание abs_max.c

```
Open  ▾  [icon]  *abs_max.c  ~/Desktop/Test/Prog2  Save  [icon]  [icon]  [icon]  [icon]
*abs_max.c  x  Makefile  x  *max.h  x

1#include "max.h"
2int abs_max(int arr[], int size){
3    int result = arr[0];
4    int max = abs(arr[0]);
5    for(int i = 1; i < size; i++){
6        if(max < abs(arr[i]))
7        {
8            max = abs(arr[i]);
9            result = arr[i];
10        }
11    }
12    return result;
13}
```

3. Создание min.h

```
1
2int abs_min(int arr[], int size);
3
```

4. Создание abs_min.c

```
1#include "min.h"
2
3int abs_min(int arr[], int size){
4    int min = abs(arr[0]);
5    int result = arr[0];
6    for(int i = 1; i < size; i++){
7        if(min > abs(arr[i]))
8        {
9            min = abs(arr[i]);
10           result = arr[i];
11        }
12    }
13    return result;
14}
```

5. Создание diff.h

```
1#include "max.h"
2#include "min.h"
3
4int diff(int arr[], int size);
```

6. Создание diff.c

```
1#include "diff.h"
2int diff(int arr[], int size)
3{
4    return(abs_max(arr, size) - abs_min(arr, size));
5}
6
```

7. Создание sum.h

```
*abs_max.c  x  Makefile  x  *max.h  x  *min.h  x  abs_min.c  x  diff.h  x  diff.c  x  sum.h  x
1int sum(int arr[], int size, int max);
```

8. Создание sum.c

```
Open  ▾  [icon]  sum.c  ~/Desktop/Test/Prog2  Save  [icon]  [icon]  [icon]  [icon]
*abs_max.c  x  Makefile  x  *max.h  x  *min.h  x  abs_min.c  x  diff.h  x  diff.c  x  sum.h  x  sum.c  x

1#include "sum.h"
2#include "min.h"
3int sum(int arr[], int size, int max)
4{
5    int start = 0;
6    int sum = 0;
7    for (int i = 0; i < size; i++)
8    {
9        if (arr[i] == max)
10        {
11            start = 1;
12        }
13        if (start)
14        {
15            sum += arr[i];
16        }
17    }
18    return sum;
19}
```

9. Создание menu.c

```
Open menu.c Save
~/Desktop/Нпо_Тн_Б2

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "diff.h"
4 #include "sum.h"
5 #define N 100
6 int main()
7 {
8     char c;
9     int array[N];
10    int choice;
11    int size = 0;
12    int max;
13    scanf("%d\n", &choice);
14    while (size <= N)
15    {
16        scanf("%d%c", &array[size], &c);
17        size++;
18        if (c == '\n')
19        {
20            break;
21        }
22        if (size < 1)
23        {
24            printf("Данные некорректны");
25            return 0;
26        }
27    }
28    switch (choice)
29    {
30        case 0:
31            printf("%d\n", abs_max(array, size));
32            break;
33        case 1:
34            printf("%d\n", abs_min(array, size));
35            break;
36        case 2:
37            printf("%d\n", diff(array, size));
38            break;
39        case 3:
40            max = abs_max(array, size);
41            printf("%d\n", sum(array, size, max));
42            break;
43        default:
44            printf("Данные некорректны\n");
45            break;
46    }
47    return 0;
48 }
49
50
51
```

10. Создание Makefile

```
1 all: abs_max.o abs_min.o diff.o sum.o menu.o
2 gcc abs_max.o abs_min.o diff.o sum.o menu.o -o menu
3 abs_max.o: abs_max.c max.h
4 gcc -c abs_max.c
5 abs_min.o: abs_min.c min.h
6 gcc -c abs_min.c
7 diff.o: diff.c diff.h
8 gcc -c diff.c
9 sum.o: sum.c sum.h
10 gcc -c sum.c
11 menu.o: menu.c
12 gcc -c menu.c
```

Выводы.

Результатом лабораторной работы является полностью рабочая программа, удовлетворяющая всем указанным в цели лабораторной работы условиям. Для сборки программы достаточно запустить команду make из папки с исходным кодом проекта. Результатом сборки проекта является исполняемый файл menu. Проверим, что программа работает правильно.

```
ngoyen@ngoyen-Vostro-3578: ~/Desktop/Ngo_Thi_lb2
gcc -c abs_max.c
gcc -c abs_min.c
gcc -c diff.c
gcc -c sum.c
gcc -c menu.c
gcc abs_max.o abs_min.o diff.o sum.o menu.o -o menu
ngoyen@ngoyen-Vostro-3578:~/Desktop/Ngo_Thi_lb2$ ./menu
0 1 2 -4 5 -8 7
-8
ngoyen@ngoyen-Vostro-3578:~/Desktop/Ngo_Thi_lb2$ ./menu
1 1 2 -4 5 -8 7
1
ngoyen@ngoyen-Vostro-3578:~/Desktop/Ngo_Thi_lb2$ ./menu
2 1 2 -4 5 -8 7
-9
ngoyen@ngoyen-Vostro-3578:~/Desktop/Ngo_Thi_lb2$ ./menu
3 1 2 -4 5 -8 7
-1
ngoyen@ngoyen-Vostro-3578:~/Desktop/Ngo_Thi_lb2$ ./menu
4 1 2 -4 5 -8 7
Данные некорректны
ngoyen@ngoyen-Vostro-3578:~/Desktop/Ngo_Thi_lb2$
```