

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Программирование»
Тема: Сборка программ в Си

Студент гр. 0382		Шангичев В. А.
Преподаватель		Жангиров Т. Р.

Санкт-Петербург
2020

Цель работы.

Скомпилировать файлы проекта, используя Makefile.

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 число в массиве. (`max.c`)
- 1 : минимальное число в массиве. (`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".

Основные теоретические положения.

В программе использовались следующие конструкции:

- Стандартная библиотека Си `studio.h`;
- Именованная константа, объявленная с помощью синтаксической конструкции `#define`;
- Функции `max`, `min`, `diff` и `main`;
- Для вывода и ввода данных использовались функции `scanf` и `printf`;
- Также для хранения данных в программе были использованы переменные и целочисленный массив;
- Логические конструкции `switch`, `else`, `if`;
- `break`;
- Циклы `while` и `for`;
- Команда `gcc <имена файлов, необходимых для компиляции>`;
- Флаг `-o`, позволяющий задать имя компилируемого файла

Выполнение работы.

1. Функция `max`.

Цель данной функции получать на вход массив и его размер и возвращать максимальное число в массиве. Сначала объявляются переменные `i` и `maximum`. Переменная `i` будет использоваться для обращения к элементам массива по индексу, а `maximum` - для хранения текущего максимального значения. Далее используется цикл `for`, где перебираются все элементы массива и выбирается максимум, который и будет возвращен в конце выполнения данной функции. Прототип данной функции расположен в заголовочном файле `max.h`. Описание – в файле `max.c`

2. Функция `min`.

Данная функция действует по такому же алгоритму, как и функция `max` (см. выше), но её задача заключается в нахождении минимального элемента массива. Существенным отличием будет лишь проверка в цикле `for`. Прототип данной функции расположен в заголовочном файле `min.h`. Описание – в файле `min.c`

3. Функция `diff`.

Эта функция должна выводить разницу между максимальным и минимальным элементом в массиве. Для нахождения максимума и минимума вызываются описанные выше функции `min` и `max`. После этого вычисляется разность этих значений и возвращается в качестве результата. Прототип данной функции расположен в заголовочном файле `diff.h`. Описание – в файле `diff.c`. Также в файле `diff.c` присутствует включение заголовочных файлов `max.h` и `min.h`

4. Функция `sum`.

В данной функции мы считаем сумму элементов массива, расположенных до первого

минимального элемента. Сначала мы вычисляем минимум, используя функцию `min`. Значение суммы будет сохраняться в переменной `sum`. При помощи цикла `for` перебираются все элементы массива. На каждой итерации мы проверяем, не равен ли этот элемент найденному минимуму. Если нет, то значение прибавляется к сумме, и выполнение цикла продолжается. Если да, то выполнение программы прекращается с помощью возвращения значения текущей суммы. Прототип данной функции расположен в файле `sum.h`. Описание – в файле `sum.c`. Также в файле присутствует включение заголовочного файла `min.h`

5. Подключение стандартной библиотеки.

В начале главного файла программы `menu.c` подключается заголовочный файл `stdio.h`, являющийся стандартным заголовочным файлом языка Си. Также в этом файле присутствуют включения всех заголовочных файлов программы: `max.h`, `min.h`, `diff.h`, `sum.h` и объявления именованной константы `MAX_LEN` для ограничения максимальной длины массива.

6. Функция `main`.

Выполнение программы начинается с функции `main`. В начале программы объявляется несколько переменных и целочисленный массив. Затем пользователю предлагается ввести число от 1 до 4 (результат сохраняется в переменную `query`). Далее с помощью цикла `while` вводятся элементы массива. Итерации выполняются до тех пор, пока мы не превысим размер массива или символом разделения окажется не пробел. В каждой итерации в массиве сохраняется целочисленное значение, а переменной `sep` присваивается строка, идущая вслед за этим значением. Следующим шагом объявляется оператор `switch`, в котором будет отслеживаться значение переменной `query`. В каждом логическом блоке `case` вызывается нужная функция, и значение, возвращенное ей, сохраняется в переменной `answer`. После чего данное значение печатается на экран. Если же переменная не равняется ни одному из допустимых значений, то на экран выводится сообщение о том, что данные некорректны.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1	0 6 92 70 59 63 90 38 57 29 0 47 81 5 48 93 29 69 93 56 23 59 25 33 91 77 71 34 85 62 41 84 66 8 41 25 89 13 68	93	С данной задачей программа справилась успешно.

Выводы.

Удалось скомпилировать файлы проекта, используя makefile. Были созданы заголовочные файлы `max.h`, `min.h`, `sum.h` и `diff.h`. Для каждой функции был создан отдельный файл. Всё это значительно упростило дальнейшую работу с проектом.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл menu.c (исходный код файла хранится в .../src/menu.c)

```
#include <stdio.h>
#include "sum.h"
#include "max.h"
#include "min.h"
#include "diff.h"
#define MAX_LEN 100

int main() {
    int query, size, answer;
    int values[MAX_LEN];
    char sep = ' ';
    size = 0;

    scanf("%d", &query);
    while (size < MAX_LEN && sep == ' '){
        scanf("%d%c", &values[size++], &sep);
    }

    switch (query){
        case 0:
            answer = max(values, size);
            printf("%d\n", answer);
            break;
        case 1:
            answer = min(values, size);
            printf("%d\n", answer);
            break;
        case 2:
            answer = diff(values, size);
            printf("%d\n", answer);
            break;
        case 3:
            answer = sum(values, size);
```

```

        printf("%d\n", answer);
        break;
    default:
        printf("Данные некорректны\n");
        break;
    }
    return 0;
}

```

Файл `sum.c` (исходный код файла хранится в `.../src/sum.c`)

```

#include "min.h"

int sum(int arr[], int size){
    int minimum = min(arr, size);
    int sum = 0;
    int i;
    for (i = 0; i < size; i++){
        if (arr[i] != minimum){
            sum += arr[i];
        }
        else {
            break;
        }
    }
    return sum;
}

```

Файл `max.c` (исходный код файла находится в `.../src/max.c`)

```

int max(int arr[], int size){
    int i = 0, maximum;
    maximum = arr[i];

    for (i = 1; i < size; i++){
        if (arr[i] > maximum) {
            maximum = arr[i];
        }
    }
}

```

```
    }  
    return maximum;  
}
```

Файл `min.c` (исходный код файла находится в `.../src/min.c`)

```
int min(int arr[], int size){  
    int i = 0;  
    int minimum;  
    minimum = arr[i];  
  
    for (i = 1; i < size; i++){  
        if (arr[i] < minimum) {  
            minimum = arr[i];  
        }  
    }  
    return minimum;  
}
```

Файл `diff.c` (исходный код файла находится в `.../src/diff.c`)

```
#include "max.h"  
#include "min.h"  
  
int diff(int arr[], int size){  
    int diff;  
    diff = max(arr, size) - min(arr, size);  
    return diff;  
}
```

Файл `max.h` (исходный код файла находится в `.../src/max.h`)

```
int max(int arr[], int size);
```

Файл `min.h` (исходный код файла находится в `.../src/min.h`)

```
int min(int arr[], int size);
```


Файл `diff.h` (исходный код файла находится в `.../src/diff.h`)

```
int diff(int arr[], int size);
```

Файл `sum.h` (исходный код файла находится в `.../src/sum.h`)

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

Файл `Makefile` (исходный код файла находится в `.../src/Makefile`)

```
all: menu
```

```
menu: menu.c max.c min.c diff.c sum.c sum.h diff.h min.h max.h
```

```
    gcc max.c min.c diff.c sum.c menu.c sum.h diff.h min.h max.h -o menu
```