

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

Студентка гр. 1304

Чернякова В.А.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Изучение процесса сборки программ, написанных на языке C на примере использования make-файлов.

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".

Выполнение работы.

Для достижения поставленной цели был создан проект, состоящий из пяти отдельных файлов и четырех заголовочных файлов. Для сборки программы был реализован make-файл *Makefile*.

В *menu.c* – файле, который реализует главную функцию, помимо стандартного заголовочного файла *stdio.h* указаны заголовочные файлы, в которых объявлены дополнительные функции, требуемые для работы программы: *max.h*, *min.h*, *diff.h*, *sum.h*. В теле главной функции с помощью функции *scanf()* с клавиатуры считывается одно из значений 0, 1, 2, 3 и массив данных, состоящий из целых чисел, количество которых не

превосходит 100. В зависимости от переданного значения оператору *switch* выполняется одна из одноименных заголовочным файлам функция, описанная в теле оператора. С помощью функции *printf()* выводится результат работы программы, и она завершается.

Прототипы:

В заголовочных файлах *max.h*, *min.h*, *diff.h*, *sum.h* находятся прототипы функций, реализованных в основных файлах. Для функции *min()* - *int min(int array[], int index);* , функции *max()* - *int max(int array[], int index);* , функции *diff()* - *int diff(int array[], int index);* , функции *sum()* - *int sum(int array[], int index);*. Каждый из прототипов содержит описание того, что приходит на вход в качестве аргументов: массив целых чисел *array[]* и целочисленная переменная *index*, в которой хранится значение длины массива.

Функции:

В файлах *max.c*, *min.c*, *diff.c*, *sum.c* описаны функции, которые используются в файле *menu.c* при реализации работы оператора *switch*.

В файле *max.c* объявлен одноименный заголовочный файл, который содержит прототип данной функции. Далее реализуется функция, которая возвращает значение максимального элемента массива *return maximum*.

В файле *min.c* объявлен одноименный заголовочный файл, который содержит прототип данной функции. Далее реализуется функция, которая возвращает значение минимального элемента массива *return minimum*.

В файле *diff.c* объявлен одноименный заголовочный файл, который содержит прототип данной функции. Далее реализуется функция, которая возвращает числовое значение разницы между самым большим и самым маленьким элементом массива *return difference*.

В файле *sum.c* объявлен одноименный заголовочный файл, который содержит прототип данной функции. Далее реализуется функция, которая возвращает значение суммы элементов массива, расположенных до первого минимального элемента этого массива *return summa*.

Make-файл:

В make-файле, который называется *Makefile*, описаны правила сборки приложения. Основной целью данного make-файла является цель *all*, в которой объявлены следующие зависимости *menu.o max.o min.o diff.o sum.o*.

Далее каждая из описанных ранее зависимостей становится самостоятельной целью, которая имеет свои зависимости и команды.

Зависимостью целей, описанных ранее, являются одноименные файлы с расширением *.c* и заголовочные файлы, включённые в него. Командами же является получение объектного файла, компилируя одноименный файл с расширением *.c* с помощью команды *gcc -c*.

В основной цели одной из команд является создание бинарного файла *menu* с помощью команды *-o*.

В конце make-файла описана цель *clean* со следующей командой *rm -rf *.o*, которая удаляет в папке все объектные файлы.

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 -21 10 0 -23 -7 -15 -14 8 -9 10 -13 -14 -27 0 -7 12 18	18	Вызывается блок с функцией <i>int max()</i> , которая возвращает значение максимального элемента массива <i>arr</i> .
2.	1 -8 -23 -30 -11 -28 15 -20 -24 -27 5 -13 5 21 -5 16 30 -12 15 -14 -28 -27 -11 -5 4 29 -5	-30	Вызывается блок с функцией <i>int min()</i> , которая возвращает значение минимального элемента массива <i>arr</i> .
3.	2 1 16 2 -18 -22 15 -3 13 0 -6 1 9 24 1 -18 15 28 20 -17 16 -11	50	Вызывается блок с функцией <i>int diff()</i> , которая возвращает

			значение разницы между максимальным элементом массива <i>arr</i> и его минимальным элементом.
4.	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	0	Вызывается блок с функцией <i>int sum()</i> , которая возвращает значение суммы элементов массива <i>arr</i> , расположенных до первого минимального элемента этого же массива.
5.	12 -5 -3 -5 -8 3 -9 -3	Данные некорректны	Вызывается блок <i>default</i> , так как ни одно из значений ни совпало.

Выводы.

Были освоены сборки программы на языке СИ, освоена утилита *make* и новые расширения: *.o* и *.h*. Создан проект состоящий из пяти отдельных файлов и 4 заголовочных, сборка которого была реализована с помощью *make*-файла *Makefile*.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: menu.c

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

int main(){
    int arr[100], ind = 0, n = 0;
    char symb;
    scanf ("%d", &n);
    while (symb != '\n'){
        scanf ("%d%c", &arr[ind], &symb);
        ind++;
    }
    switch (n){
        case 0:
            printf ("%d\n", max(arr, ind));
            break;
        case 1:
            printf ("%d\n", min(arr, ind));
            break;
        case 2:
            printf ("%d\n", diff(arr, ind));
            break;
        case 3:
            printf ("%d\n", sum(arr, ind));
            break;
        default:
            printf ("Данные некорректны\n");
            break;
    }
}
```

Название файла: min.h

```
int min(int array[], int index);
```

Название файла: min.c

```
#include "min.h"
```

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

Название файла: max.h

```
int max(int array[], int index);
```

Название файла: max.c

```
#include "max.h"
```

```
int max(int array[], int index){  
    int maximum = array[0], i = 0;  
    for (i; i < index; i++)  
        if (array[i] > maximum)  
            maximum = array[i];  
    return maximum;  
}
```

Название файла: diff.h

```
int diff(int array[], int index);
```

Название файла: diff.c

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

```
#include "max.h"
```

```
#include "min.h"
```

```
int diff(int array[], int index){
```

```

        int difference;
        difference = max(array, index) - min(array, index);
        return difference;
    }

```

Название файла: sum.h

```
int sum(int array[], int index);
```

Название файла: sum.c

```

#include "sum.h"
#include "min.h"

int sum(int array[], int index){
    int summa = 0, i = 0;
    while (array[i] != min(array, index)){
        summa = summa + array[i];
        i++;
    }
    return summa;
}

```

Название файла: Makefile

```

all: menu.o max.o min.o diff.o sum.o
    gcc menu.o max.o min.o diff.o sum.o -o menu

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

min.o: min.c min.h
    gcc -c min.c

diff.o: diff.c diff.h max.h min.h
    gcc -c diff.c

sum.o: sum.c sum.h min.h

```



```
gcc -c sum.c
```

```
clean:
```

```
rm -r *.o
```