

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
Вариант 2

Студент гр. 0382

Афанасьев Н. С.

Преподаватели

Чайка К. В.,
Жангиров Т. Р

Санкт-Петербург

2020

Цель работы.

Изучение процесса сборки программ в языке C.

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 начала, в теле функции `main()` (файл `menu.c`) создаются три переменные: `array` – массив на 100 элементов, в котором будут храниться числа, `len` – количество введенных чисел и `operation` – номером операции, первым числом во входной строке.

Для ввода чисел с клавиатуры используется цикл `while`: переменная `len` здесь используется как счётчик и каждый раз увеличивается на единицу; с помощью метода `scanf()` считывается следующее число и записывается в массив `array` под индексом `len`; через метод `getchar()` считывается следующий за числом символ, и если он равен символу перевода строки `'\n'`, то работа цикла завершается.

Для того, чтобы понять, какую операцию использовать, в теле функции *main()* используется оператор *switch*, который принимает номер операции в качестве аргумента и в зависимости от операции выводит её результат. Если такой номер не найден, выводит строку "Данные некорректны".

Для каждой операции создаётся свой файл с функцией и заголовочный файл, название файла совпадает с названием из условия. В качестве аргументов все функции принимают начальный массив *array* и количество чисел *len*. В теле этих функций также используются переменная *res* для хранения результата функции и *j* в качестве счётчика для цикла *for*.

- Функции *max()* и *min()* (файлы *max.c*, *min.c*) проходят весь массив через цикл *for*, ищут максимальное/минимальное значение и возвращают его индекс в массиве.
- Функция *diff()* (файл *diff.c*) возвращает разность между максимальным и минимальным значением, полученных с помощью функций *max()* и *min()*.
- Функция *sum()* (файл *sum.c*) складывает значения всех элементов до индекса с первым минимальным элементом, полученным через *min()*.

В случае успешного выполнения программы функция *main()* возвращает значение 0.

В файле *Makefile* содержатся инструкции по сборке проекта: для начала из исходных файлов (*menu.c* и все функции в папке *assets*) создаются объектные файлы (*.o*) и помещаются в папку *obj*. Потом эти объектные файлы используются для сборки исполняемого файла *menu*. Также *Makefile* содержит инструкцию *clear* (*clear-w* для *windows*), которая удаляет 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 6 92 -70 59 63 90 57 29 0 - 69 93 56 -23 59 25 33 -100	93	Верно, максимальное значение – 93
2.	1 6 92 -70 59 63 90 57 29 0 - 69 93 56 -23 59 25 33 -100	-100	Верно, минимальное значение – -100
3.	2 6 92 -70 59 63 90 57 29 0 - 69 93 56 -23 59 25 33 -100	193	Верно, разница между максимальным и минимальным значением – $93 - (-100) = 193$
4	3 10 0 10 30 10 40 -100 100 93 -100 0 -5 120 -100 18 -57	100	Верно, сумма чисел до первого минимального значения $(-100) - 10*3 + 0 + 30 + 40 = 100$
5	-94 6 92 -70 59 63 90 57 29 0 -69 93 56 -23 59 25 33 -10	"Данные некорректны"	Верно, операции под номером -94 не существует

Выводы.

Был изучен процесс сборки программ в языке C с помощью утилиты *make*.

Разработана программа, выполняющая считывание с клавиатуры номера операции и массива чисел с помощью цикла *while*, определяющая, какую операцию необходимо выполнить через оператор *switch*, выполняющая расчёт результата в отдельной функции, принимающей в качестве аргументов исходный массив и количество введенных чисел, и выводящая этот результат пользователю.

Все функции хранятся в отдельных файлах. Созданный *make*-файл содержит инструкции по сборке программы и очистке объектных файлов. В результате сборки создаётся исполняемый файл *menu*.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

1. Название файла: menu.c

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

int main(){
    int len = 0, array[100], operation;

    scanf("%d", &operation);

    while(scanf("%d", &array[len])){
        len++;
        if(getchar() == '\n') break;
    }

    switch(operation){
        case 0:
            printf("%d\n", array[max(array, len)]);
            break;
        case 1:
            printf("%d\n ", array[min(array, len)]);
            break;
        case 2:
            printf("%d\n ", diff(array, len));
            break;
        case 3:
            printf("%d\n ", sum(array, len));
            break;
        default:
            printf("Данные некорректны\n ");
    }
    return 0;
}
```

2. Название файла: max.h

```
int max(int *array, int len);
```

3. Название файла: max.c

```
#include "max.h"

int max(int *array, int len){
    int res = 0;
    for(int j = 0; j < len; j++){
        if(array[j] > array[res]) res = j;
    }
    return res;
}
```

4. Название файла: min.h

```
int min(int *array, int len);
```

5. Название файла: min.c

```
#include "min.h"

int min(int *array, int len){
    int res = 0;
    for(int j = 0; j < len; j++)
        if(array[j] < array[res]) res = j;
    return res;
}
```

6. Название файла: diff.h

```
int diff(int *array, int len);
```

7. Название файла: diff.c

```
#include "diff.h"
#include "max.h"
#include "min.h"

int diff(int *array, int len){
    return array[max(array, len)] - array[min(array, len)];
}
```

8. Название файла: sum.h

```
int sum(int *array, int len);
```

9. Название файла: sum.c

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

int sum(int *array, int len){
    int res = 0;
    for(int j = 0; j < min(array, len); j++)
        res += array[j];
    return res;
}
```

10. Название файла: Makefile

```
all: obj obj/menu.o obj/max.o obj/min.o obj/diff.o obj/sum.o
    gcc obj/menu.o obj/max.o obj/min.o obj/diff.o obj/sum.o -o menu
obj:
    mkdir obj
obj/menu.o: menu.c
    gcc -c menu.c -o obj/menu.o
obj/max.o: assets/max.c
    gcc -c assets/max.c -o obj/max.o
obj/min.o: assets/min.c
    gcc -c assets/min.c -o obj/min.o
obj/diff.o: assets/diff.c
    gcc -c assets/diff.c -o obj/diff.o
obj/sum.o: assets/sum.c
    gcc -c assets/sum.c -o obj/sum.o
clear:
    rm -r obj
clear-w:
    rd obj /s/q
```