

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 С».

Студент гр. 1304

Байков Е.С.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы

Написать программу в соответствии с условием задачи, а также написать make-файл.

Задание

Вариант 2

Напишите программу, выделив каждую подзадачу в отдельную функцию.

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

В программе реализованы файлы: *max.c*, *min.c*, *diff.c*, *sum.c*, *menu.c*, *max.h*, *min.h*, *diff.h*, *sum.h*. В файлах *max.h*, *min.h*, *diff.h*, *sum.h* содержится прототип каждой функции, описанной в *max.c*, *min.c*, *diff.c*, *sum.c*

соответственно. На вход каждой функции подается *array[]*(массив, считываемый с клавиатуры) и *len*(количество элементов, введенных в массив).

max.c: инициализируется переменная *maximum*, которой передается значение *array[0]*. После с помощью цикла *for(int i = 1; i < len; i++)* и условного оператора *if* каждый элемент массива по порядку сравнивается с текущим значением переменной *maximum* и записывает новое значение в данную переменную, если значение элемента в массиве больше *maximum*. Возвращает *maximum*.

min.c: инициализируется переменная *minimum*, которой передается значение *array[0]*. После с помощью цикла *for(int i = 1; i < len; i++)* и условного оператора *if* каждый элемент массива по порядку сравнивается с текущим значением переменной *minimum* и записывает новое значение в данную переменную, если значение элемента в массиве меньше *minimum*. Возвращает *minimum*.

diff.c: функция использует вышеописанные функции и возвращает разность их возвращаемых значений.

sum.c: функция использует функцию *min* и записывает значение возвращаемое функцией в переменную *minimum*, затем инициализируется переменная *total*, которой присваивается значение 0. С помощью цикла *for(int i = 0; i < len; i++)* и условного оператора *if* программа прибавляет к значению *total* значение *array[i]*, пока не встретит первое значение равное *minimum*. Возвращает *total*.

menu.c: Объявление и описание функции *main*, содержащей переменные типа *int*: *command*, *array[100]*, *len* – а также типа *char*: *sign*. Переменной *len* присваивается значение 0.

Внутри функции *main* с помощью функции *scanf* со спецификаторами *%d%c* происходит считывание значения, которое потом заносится в переменные *command* и *sign* соответственно. С помощью цикла *for* и вызова функции *scanf* с теми же спецификаторами внутри него в массив *array[100]* записываются значения, поданные на вход, а переменной *sign* присваивается

значение знака после (пробел или знак переноса строки) числа. Идет увеличение значения переменной *len* на единицу за каждую итерацию. После того как значение переменной *sign* становится равно символу переноса строки с помощью условного оператора *if* и оператора *break* происходит досрочное завершение цикла. Вводится оператор *switch* который принимает значение команды (*command*). В блоках *case* обрабатывается значение, поданное на вход, и с помощью функции *printf* выводится значение соответствующей функции, которой на вход были поданы массив *array* и переменная *len*, или же, если введен номер программы, которой не существует, в блоке *default* с помощью функции *printf* выводится строка “\nДанные некорректны”. В конце функция возвращает 0.

Makefile: В файле содержится 7 целей: *all*, *menu.o*, *max.o*, *min.o*, *diff.o*, *sum.o*, *clean*.

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 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	Возвращает максимальное значение
2.	1 93 45 34 34 56 324 44 5 23 12 34 0 5 65 65 78 89 31 43 55 67 45 342 112 56	0	Возвращает минимальное значение
3.	2 100 13 15 34 43 65 56 43 21 90 44 56 76 77 84 93 41 66 10	90	Возвращает разность между максимумом и минимумом
4.	3 44 56 72 28 44 56 30 40 30 10 23 43 54 66 77 100 32 100 23 45 56 89	400	Возвращает сумму чисел до первого минимума
5.	22 40 34 54 77 89 123 44 55 67 89 90 33 43 56 76 34 23 10 11 14 56 99 93 34 54	Данные некорректны	Ввод некорректной команды и вывод надписи: «Данные некорректны»

Выводы

Были изучены работа препроцессора, компилятора и компоновщика.

Программа, обрабатывающая введенные данные согласно указанному алгоритму, была разделена на несколько файлов. Написан *Makefile*, с помощью которого создается исполняемый файл путем компоновки объектных файлов, полученных в результате компиляции соответствующих исходных файлов.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

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
```

Название файла: **menu.c**

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

int main(){
    int command, array[100], len = 0;
    char sign;
    scanf("%d%c", &command, &sign);
    for (int i = 0; i < 100; i++){
        scanf("%d%c", &array[i], &sign);
        len += 1;
        if(sign == '\n'){
            break;
        }
    }
    switch(command){
        case 0:
            printf("\n%d", max(array, len));
            break;
        case 1:
            printf("\n%d", min(array, len));
            break;
        case 2:
            printf("\n%d", diff(array, len));
            break;
        case 3:
            printf("\n%d", sum(array, len));
```

```

        break;
    default:
        printf("\nДанные некорректны");
    }
    return 0;
}

```

Название файла: **max.h**

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

Название файла: **max.c**

```

#include "max.h"

int max(int array[], int len){
    int maximum = array[0];
    for (int i = 1; i < len; i++){
        if (array[i] > maximum){
            maximum = array[i];
        }
    }
    return maximum;
}

```

Название файла: **min.h**

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

Название файла: **min.c**

```

#include "min.h"

int max(int array[], int len){
    int minimum = array[0];
    for (int i = 1; i < len; i++){
        if (array[i] < minimum){
            minimum = array[i];
        }
    }
    return minimum;
}

```

Название файла: **diff.h**

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

Название файла: **diff.c**

```

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

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

```

Название файла: **sum.h**

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

Название файла: **sum.c**

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

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

```
int sum(int array[], int len){
    int minimum = min(array, len);
    int total = 0;
    for(int i = 0; i < len; i++){
        if (array[i] != minimum){
            total += array[i];
        }
        else{
            break;
        }
    }
    return total;
}
```