

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. 0382

Салихов М.Ю.

Преподаватель

Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы.

Изучение препроцессора, утилиты Make.

Задание.

Вариант 2

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 программу, 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)

1 : минимальное число в массиве. (min)

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

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

Иначе необходимо вывести строку "Данные некорректны".

Основные теоретические положения.

При написании программы были использованы:

Стандартная библиотека `<stdio.h>`

Стандартные функции ввода и вывода `scanf()` и `printf()` соответственно.

Директива компиляции Header guard:

`#ifndef X`

`...`

`#endif`

Циклы:

`for(<>; <>; <>){}` - в круглых скобках первым аргументом объявляется инициализация счетчика, вторым аргументом - условие цикла, третьим — наращивание счетчика.

`while(){}` - если выражение в круглых скобках истинно, выполняется блок кода, затем снова проверяется условие цикла, и начинается следующая итерация.

Операторы:

`if(){}` - если условие истинно, выполняется блок. Если далее присутствует опция `else{}`, то выполняются выражения следующие за ней в фигурных скобках. (Опция `else` в данной программе не использовалась)

`switch(){case a:, case b:, ... case n:, default:}` - значение переменной сравнивается со значениями, которые следуют за ключевым словом `case`, и при совпадении выполняются соответствующие данному `case` код и следующие за ним под-`case`-овые выражения. При отсутствии совпадений, выполняется код при `default`.

Выполнение работы.

Сначала подключается стандартная библиотека `<stdio.h>`, затем определяется имя `VOLUME`(размер максимально допустимого массива) как `101`. Нулевой элемент массива выделяется для «ключевого» значения (для выбора опции в `switch`). Еще сто необходимо из условия задачи. Еще ниже прототипируются собственные функции `max`, `min`, `diff` и `sum`.

Функция `main`: инициализация переменных:

`int Numbers[VOLUME]` — массив из 101 элемента.

char Symb — в эту переменную помещается символ пробела или *ENTER*.

int VLM — сюда помещается кол-во чисел, полученных от пользователя (не включая «ключа»).

С помощью функции *scanf* и цикла *for* считывается строка чисел. При помощи оператора *if* на каждой итерации цикла проверяется, является ли полученный символ - символом перевода строки.

Оператор *switch*: принимает значение нулевого индекса массива (первое напечатанное пользователем число). В *case 0:*, *case 1:*, *case 2:* и *case 3:* помещено по функции *printf*, печатающие: если (0), то максимальное число массива, (1)минимальное, (2)разницу между ними и (3)сумму чисел строки, встреченных слева на право до минимального числа массива. Если «ключ» не равняется 0, 1, 2 или 3, то через оператор *default* и функцию *printf* выводится строка «Данные некорректны».

В случае успешного завершения работы программы, с помощью оператора *return* возвращает значение 0.

Собственные функции:

Все четыре нижеописанные функции принимают количество чисел(*int n*) в массиве и сам массив(*int Arr[]*)(сами функции не оперируют с нулевым элементом массива). Возвращают функции высчитанные значения с помощью оператора *return*.

int max(int n, int Arr[]) - с помощью цикла *for* и оператора *if* проверяет на каждой новой итерации цикла, является ли последующий элемент массива больше предыдущих. Если больше, то в переменную *int maxi* записывается это новое число. В начале функции, при инициализации переменных, по умолчанию максимальным элементом считается первый (*int maxi = 1*). Возвращает индекс максимального элемента в массиве.

int min(int n, int Arr[]) - работает по аналогии с функцией *max*. Возвращает индекс минимального элемента.

int diff(int n, int Arr[]) - функция использует при расчете функции *min* и *max*. При помощи стандартного унарного знака «-» находит разницу между максимальным и минимальным числами массива. Возвращает эту разницу.

int sum(int n, int Arr[]) - при работе использует функцию *min*. Инициализируются переменные *int sumUN = 0* и (в самом цикле *for*) счетчик *int i = 1*. С помощью цикла *for* суммирует числа с первого по первый встреченный минимальный элемент массива. Возвращает сумму.

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.	1 2 3 4 5 6	2	Соответствует задаче программы.
2.	3 213 -421 56 72 123 -3 23 18	213	Соответствует задаче программы.
3.	-16 43 2 86 7 12 34	Данные некорректны	Соответствует задаче программы.
4	2 45 6 18 2 90 -4	94	Соответствует задаче программы.
5.	0 1 2 332 -98 23 44 1	332	Соответствует задаче программы.

Выводы.

Было изучены базовые конструкции языка Си.

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. С помощью функций *for*, *scanf* и *if* была принята и обработана (отсортирована) строка чисел, пробелов и знака перевода строки. Затем, с помощью оператора *switch* была выбрана необходимая, исходя из значения первого принятого числа строки, нужная пользователю функция. А, наконец, функция *printf* вывела результат вычислений программы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: menu.c

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

#define VOLUME 101

int main(){

    int Numbers[VOLUME];
    char Symb;
    int VLM;

    for (int i = 0; i < VOLUME; i++){
        scanf("%d%c", &Numbers[i], &Symb);
        if (Symb == 10){
            VLM = i;
            break;}
    }

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

Название файла: max.c

```
#include "max.h"

//f() поиск номера первого макс. элемента

int max(int n, int Arr[]){
    int maxi = 1;
    int i;
    for(i = 2; i <= n; i++){
        if (Arr[i] > Arr[maxi]) maxi = i;
    }
    return maxi;
}
```

Название файла: max.h

```
#ifndef MAX_H
#define MAX_H
int max(int, int *);
#endif
```

Название файла: min.c

```
#include "min.h"

//f() поиск номера первого мин. элемента

int min(int n, int Arr[]){
    int mini = 1;
    int i;
    for(i = 2; i <= n; i++){
        if (Arr[i] < Arr[mini]) mini = i;
    }
    return mini;
}
```

Название файла: min.h

```
#ifndef SUM_H
#define SUM_H
int sum(int, int *);
#endif
Название файла: diff.c
#include "diff.h"
#include "max.h"
#include "min.h"
```

```
//f() разница между макс. и мин. элементами

int diff(int n, int Arr[]){
    return (Arr[max(n, Arr)]-Arr[min(n, Arr)]);
}
```

Название файла: diff.h

```
#ifndef DIFF_H
#define DIFF_H
int diff(int, int *);
#endif
```

Название файла: sum.c

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

//f() сумма всех встреченных элементов до первого мин. элемента в массиве

```
int sum(int n, int Arr[]){
    int sumUN = 0;
    int i;
    for (i = 1; i < min(n, Arr); i++){
        sumUN += Arr[i];
    }
    return sumUN;
}
```

Название файла: sum.h

```
#ifndef SUM_H
#define SUM_H
int sum(int, int *);
#endif
```

Название файла: Makefile

```
menu: 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 max.h min.h diff.h
    gcc -c diff.c
sum.o: min.h sum.c sum.h
    gcc -c sum.c
clean:
    rm -rf *.o menu
```