CSCV 352 – Fall 2023

Exercise 2

For the assignment this week you will be turning in one source file (.c file) for each of the programs described below. Note that there is a file naming convention specified in these instructions: the file turned in for each assignment MUST be named Exercise2_*n*.c where *n* is the assignment number.

Also, there is a makefile provided that will compile your program for you IF you name your source file according to the convention (Exercise2_*n*.c). One easy way to manage the exercises is to create a folder for each part of the assignment and place a copy of the makefile in each folder. Then write your program according to the description using a source file called "Exercise2_*n*.c". You can build your program using "make", and then turn in the .c file when you are ready.

Exercise2_1

Write a program that produces the following output:

```
U2, R.E.M., Counting Crows, Matchbox Twenty
Counting Crows, Matchbox Twenty
U2, R.E.M.
```

Have the program use two user-defined functions in addition to main(): one named eighties() that prints "U2, R.E.M." once, and one named nineties() that prints "Counting Crows, Matchbox Twenty" once. **The band group names must be printed within the respective function**. Let main() control the printing process and take care of any additional printing or formatting tasks.

Turn your program in as Exercise2_1.c.

Exercise2_2

Write a program that requests two integer values from the user. The program then calculates c in the following formula:

$$c = a^3 + b$$

where a and b are the two numbers that the user entered. The program must print to the console the calculated value of c.

The program must also include error checking. If the user does not enter a valid integer, the program must print a message for the user and exit without performing the calculation.

Turn your program in as Exercise2_2.c.

Exercise2_3

Write a program that asks the user to enter a height in centimeters and then displays the height in centimeters and in feet and inches. Fractional centimeters and inches should be allowed, and the program should allow the user to continue entering heights until a nonpositive value is entered. Assume 2.54 centimeters to an inch. A sample run should look like this:

```
Enter a height in centimeters (<=0 to quit): 182
182.0 cm = 5 feet, 11.7 inches

Enter a height in centimeters (<=0 to quit): 168.7
168.7 cm = 5 feet, 6.4 inches

Enter a height in centimeters (<=0 to quit): 0
bye
```

Turn your program in as Exercise2_3.c.

Exercise2_4

Write a program that creates two seven-element arrays of type `double` and uses a loop to let the user enter values for each of the seven elements of the first array.

Have the program set the elements of the second array to the cumulative totals of the elements of the first array. For example, the fourth element of the second array should equal the sum of the first four elements of the first array, and the fifth element of the second array should equal the sum of the first five elements of the first array. (It's possible to do this with either nested or multiple loops, but by using the fact that the fifth element of the second array equals the fourth element of the second array plus the fifth element of the first array, you can avoid multiple loops and just use a single loop for this task.)

Finally, use two loops to display the contents of the two arrays, with the first array displayed on one line and with each element of the second array displayed below the corresponding element of the first array.

Turn your program in as Exercise2_4.c.

Exercise2_5

Write a program that helps the user manage their budget and pay their rent. The program prompts the user for income information and for certain expenses. The program must request the following information:

What is your hourly wage?
How many hours per week do you work?
How much do you spend on groceries each week?
What is your monthly utility bill?
What is your monthly cell phone bill?
What other expenses do you have this month?
How much is your rent?

Use the information provided by the user to determine their monthly income and their total expenses before paying rent. If the user enters more than 40 hours in a work week, assume they are paid time and a half (1.5 times their hourly wage) for any hours over 40. Tell the user if they have enough money remaining to pay their rent. They can only pay rent if after subtracting their non-rent expenses from their income the remaining funds is greater than or equal to their rent payment amount.

Turn your program in as Exercise2_5.c

TURN IN
==========

Exercise2_1.c
Exercise2_2.c
Exercise2_3.c
Exercise2_4.c
Exercise2_5.c