

CS 271
Lab Assignment # 2

Due Tuesday, Feb 9 before 2:00 pm

Submit **coffee.c** and **triples.c** on Canvas Lab 2 Assignment.

Read Chapters 3 and 4.
Powerpoints are available on Canvas (click "Modules" and look for the chapter numbers).

1. Organize your directories (folders) for this class (see the handout).
2. Download the file data1.txt from Canvas and save it in the Lab2 directory.
3. Change working directory to the Lab2 folder. Create a program named **coffee.c**.

Problem: Monthly coffee prices are tracked over a 10 year period. The CEO of Starbucks would like to know the maximum and minimum coffee prices over that period along with the months in which the maximum and minimum prices occurred.

Input: A text file containing one line for each month. A line of data contains integers for month and year, followed by a double value for price. A line with the sentinel value -1 for month indicates the end of input data. (Note: There are no values for year and price on the last line of the file.)

Output: Display the maximum price along with the month and year in which it occurred. Display the minimum price along with the month and year in which it occurred. The numbers shown here are **not the actual values**. They're just to show how the output is to be formatted.

Minimum Price	\$ 95.67	November	2010
Maximum Price	\$294.83	May	2012

- Input terminates when the value -1 is input for month. A sentinel loop is **required**.
- Do not use arrays.
- Do not use file input or output functions. No fscanf, fgets, etc.
- Include a comment on a separate line before you begin each process (examples: declare variables, input data, calculate ..., output results, etc)

- Include header comments in the following format. This header format is expected on all programs you submit this semester.

```
// CS 271 - lab assignment #
// program_name
// purpose of the program
// written by yourname
// date written
```

4. Save the program. Compile

```
gcc coffee.c -o coffee
```

5. Debug if needed and recompile. Run and test the program.

```
./coffee
```

When you run the program this way, it will get all input from the keyboard (**you have to type the data values, then type -1 at the end of the data**).

Input Redirection Run the program again using input redirection. This will take all input from the file you specify (instead of the keyboard).

```
./coffee < data1.txt
```

View the contents of the data1.txt file in your editor and determine the expected output of the program. Compare the expected output to the actual program output. If the values don't match, debug your program or correct your hand calculations (or both).

Close the program and data file once you have the correct output.

6. Create a program named **triples.c**. Solve problem 4.27 on page 155 (This is included in the lab 2 handout.) Follow the instructions in the textbook (specifically, using brute force with a triple-nested for loop) except change the maximum value to 1000.

- Include documentation just like you did in coffee.c.
- In the output, print column headings and right-justify the numbers as shown below:

Side 1	Side 2	Hypotenuse
-----	-----	-----
3	4	5
5	12	13
...		

You should not have duplicates.
For example, if you see 3, 4, 5
you should not see 4, 3, 5.

7. Compile. Debug if needed and recompile. Run and test the program. You can check your output for this one by doing a web search on "Pythagorean Triples".

Submit coffee.c and triples.c on Canvas.