GENE 121 **Assignment 8** **Fall 2016**

Question 1 of this assignment will be completed in pairs, Question 2 will be completed individually.

For this assignment, both partners should determine all function prototypes. Decide which partner is Partner A and which partner is Partner B and complete the sections accordingly. Make sure to indicate on your assignment who is A and who is B so we know who completed which parts of the assignment.

In this assignment you will be using:

- Functions
- 2D Arrays
- Classes

Don't forget to include a signed header for your assignment, and to make sure that both students in the group have an electronic copy of the code when it is complete.

Question 1 (Dev-C++)

Chris has been experimenting with brewing coffee at home. He has been experimenting with two main variables in regards to the brewing: the temperature of the water and how coarse/fine the beans have been ground. He has been experimenting with temperatures between 91 and 99 degrees C and has ranked how coarse the beans have been ground on a 4 point scale (where 1 is very fine and 4 is very coarse). After brewing (and tasting) the cup of coffee, Chris assigns it a rating out of 10 (a 1 indicating he hated it and a 10 indicating that he loved it). He has run these experiments a total of three times for all combinations of temperature and coarseness and has stored all of this experimental data in 3 files: `coffee1.txt`, `coffee2.txt` and `coffee3.txt`. All three files are formatted the same way; samples of `coffee1.txt` and `coffee2.txt` are below:

```
coffee1.txt:              coffee2.txt:
92    4     7             96    3     8
94    2     3             91    1     1
96    3     9             93    3     6
…     …     …             …     …     …
```

The first row of coffee1.txt above represents a cup of coffee brewed at 92 degrees with a grind of 4 and Chris ranked it 7/10. The second row is a cup brewed at 94 degrees with a grind of 2 and Chris ranked it 3/10. Notice that the third row of coffee1.txt and the first row of coffee2.txt show different rankings for the same combination of temperature and grind, tested on different days.

Your program should output which combination of coarseness and temperature consistently resulted in the best cup of coffee.

(a) Describe your definition of "best coffee". Submit this description with your assignment.

(b) Describe how you are storing your data. A good method for this is to draw a table representing your array(s) and fill in the sample data from above. Submit this description with your assignment.

Do not proceed until both partners are on the same page about how the data will be stored, and what "best" means.

(c) Write a function which accepts one already opened input file stream as well as a 2D array of data and updates the array with data from the input file. **Partner A**

(d) Write a function which accepts the array(s) of your coffee data and outputs the coffee data to the console in a **well-formatted table with headings**. **Partner B**

(e) Write a function which accepts the array(s) of your coffee data, a Boolean indicating rows or columns, and the index of a row/column. This function will sum all values in the indicated row/column. **Partner A**

(f) Write a function which accepts the array(s) of your aggregate (i.e. combined) coffee data. Your function should return the coarseness and temperature which consistently resulted in the best cup of coffee. **Partner B**

(g) Write a main program that: **Partner A and B**

- Opens the 3 files. There is no need to verify if they opened correctly.
- Declares appropriately sized and named 2D array(s) where you will store the coffee data
- Reads each of the 3 files and updates your array(s) using the function from (c)
- Outputs your aggregate coffee data to the console using the function from (d)
- Outputs the temperature which brewed the best coffee. Use your function from (e)
- Outputs the ideal combination of temperature and coarseness using your function from (f)

Submit your code and output.

Question 2 (Dev-C++) – To be completed individually

As we have seen in the past, some rational numbers are not stored well in binary (e.g. 1/10 is an infinitely repeating binary number). To solve this issue, we are going to create a proper `Fraction` class which stores the numerator and denominator separately. Proper fractions, by definition, have a value between -1 and 1, so you may assume that $|num| < |denom|$.

a) Write a class which will store a proper `Fraction`. The **private** data members are the **integer** numerator and denominator. This class should have the following functions:
- A default constructor (hint: What is a reasonable proper fraction that will not break later functions?)
- A data constructor (hint: What if the denominator comes in negative?)
  - Note: You do not need to worry about reducing the fraction.
- All necessary accessors and mutators
  - What checks are needed to maintain a proper `Fraction`?
- A function which accepts an integer indicating the number of decimal places, and returns an approximation of the fraction to the indicated number of decimal places
  - Hint: Ask Professor Google about the `round()` function
- A boolean function which accepts another proper `Fraction` and returns whether they are exactly the same. Use the following function header:
  - `bool isExactlySame(Fraction const & other) const{}`
- An output function which outputs to the console

b) Write a main which suitably tests your proper `Fraction` class.

Since it's your first class you will probably run into syntax errors.
The ';' missing from the end of the class can produce over 50 messages!
TA's are available to help you de-bug. Please don't spend too long troubleshooting.

Submit your code with output.