CS 560 Project 1 Due: February 23, 2016

**Description:** This project is to implement a fraction class. A fraction is represented by a pair of integers, the numerator and denominator. They may be either positive or negative, and the numerator, but not the denominator, may be zero. If a fraction is negative, it should be stored as a negative numerator and positive denominator, even if the input has the negative sign in the denominator. To reduce fractions you must use Euclid's algorithm to find the GCD.

Euclid's Algorithm:

if m divides evenly into n then the GCD(n,m) = m

else GCD(n, m) = GCD(m, n mod m)

**Specifications:** The following fraction operations are to be implemented and should produce the indicated results. All answers must be given in lowest terms. F, F1, and F2 are fractions, N, N1 and N2 are integers and R is a real number.

ADD(F1, F2) Find the sum of F1 and F2 and report the result as a fraction

XADD(F1, F2) Find the sum of F1 and F2 and report the result as a mixed number if the result is an improper fraction

SUB(F1, F2) Subtract F2 from F1

MUL(F1, F2) Find the product of the two fractions and report the result as a fraction.

DIV(F1, F2) Divide F1 by F2 and report the result as a fraction

XDIV(F1, F2) Divide F1 by F2 and report the result as a mixed number if the result is an improper fraction

REC(F) Find the reciprocal of F i.e. invert F

RED(F) Reduce F to lowest terms

D2F(R) Convert the decimal to a fraction in lowest terms

MIX(F) Change an improper fraction to a mixed number

UNM(N, F) Change a mixed number to a fraction--N is the integer part and F is the fractional part of the mixed number

AMIX(N1, F1, N2, F2) Add two mixed numbers and report the answer as a mixed number

LESS(F1, F2) Determine which fraction is smaller

You also must write the functions below that use the fraction class. These functions are not part of the fraction class and must not use any knowledge of how the fraction operations are implemented. Only the number and types of arguments for each operation are known. For all functions, the first integer in the input is N and the remaining inputs are the numerators and denominators of the fractions.

SUM function: Input: an integer N and a list of N fractions.

Output: The sum of the fractions.

When you output the result you do not need to echo all of the fractions, just report the value of N and the sum.

SORT function: Input: an integer N and a list of N fractions.

Output: The list of fractions sorted in increasing order.

MEAN function: Input: an integer N and a list of N fractions.

Output The number of fractions and their average.

MEDIAN function: Input: an integer N and a list of N fractions.

Output The mean of the fractions.

If the number of fractions is odd, the median is the middle value. If the number of fractions is even, the median is the average of the two “middle” values.

MODE function: Input: an integer N and a list of N fractions.

Output: The mode of the fractions.

The mode is the most frequently occurring value. If there is more than one node, report them all. For example, in the list 1, 2, 3, 1, 2, 3, 1 the mode is 1. In the list 1, 2, 3, 4, 1, 2 both 1 and 2 are modes.

**Miscellaneous**

1. For each command, generate an output reporting the action taken by your program. The output should indicate the operation, the operands, and the answer, if appropriate. For example, 1/2 + 1/8 = 5/8. Error messages such as illegal input, attempted division by 0, etc. should be output as appropriate. Negative fractions should be output with the negative sign in the numerator.

2. You may not convert fractions to decimals in order to perform the operations. (Besides, a fraction like 1/3 cannot be exactly represented in the computer.)

3. To make the grading a bit easier, please use the names like proj1.h and proj1.c for your program parts. Also, include a user prompt for the data file name, or include it on the command line when compiling the program.

4. Each line of the data file has the following format: Operation Arg1 Arg2 ...

The correct number of arguments will be given so you don’t have to check for that. Arguments are separated by spaces. A fraction is input as a pair of integers. The line in the data file for the example in number 1 would be: ADD 1 2 1 8

**To turn in:**

1. Electronically submit (to both the TA and to me) your well-documented code and the output produced by your program on the data file proj1.data on the web page. Include copies of any additional data files you used and the output produced by your program on those files. Please be sure everything is clearly labeled and bundled together into a zip file. Be sure to include a README file with explicit instructions on how to compile and run your program. Use a makefile. Modifying your file after sending it may cost you bonus points (or incur penalties).

2. Design your own test data for the functions that sum and sort lists of fractions and report the mean, median and mode. Part of your grade will be based on how thoroughly you tested these.

3. Include in the zip file you submit a written report that describes how your program was structured, problems encountered, etc. Also include the data files you designed for testing the various operations and functions and the results of running your program on those files.