Nathan Bryant

Andrew Han

PA5 README

For this assignment, we created three files: Rational.cpp, Rational.h and tester.cpp. Rational.cpp handles the defining of the operator overloaded functions for the Rational class, as well as a few other methods that we needed, such as getters and the normalize function. For our normalize function, we referred to Euclid’s algorithm. We also have a normalize functions for long long integers because we needed this when we were creating the new Rational after performing an operation on two rationals. Rational.h is the header file for Rational.cpp and is included by tester.cpp, and it defines the Rational class and handles other inclusions. All of the operator overload functions are member functions in our program, except for the >> and << operators, because these needed to be friend functions.

The file tester.cpp contains the main function, which handles reading the input files and outputting the results. We used a for loop that iterated when it completed an input file, and we used an inner while loop to handle reading the input files and performing the given operations. It prints the line as it reads it in to the standard output, and then prints the new rational on the line as well as a conversion to double, using the toDouble function in the Rational class.

Test Cases

For the operators == and != we have five tests each: one that was not true, one that was true, one that dealt with negative numbers, one that was true with integers and one that was not true with integers. For the other conditional operators, we have seven tests each: one that was true, one that was not true, one where the two rationals were equal, one with negative numbers and three tests with integers (one true, one false, one equal). For the arithmetic operators, we have one with relatively prime values, non-relatively prime values, negatives, dealing with integers, negatives in the denominator, having the numerator and denominator negative, and only with integers. For each of these, we also tested for divide by zero, which produces an error for that line and continues reading the file. Finally, we have three tests for multiple operations on the same line. These tests are across multiple input files, in order to test our program for reading multiple input files.

Test Cases:

Arguments:  
awhan\_nmbryant\_1.txt awhan\_nmbryant\_2.txt awhan\_nmbryant\_3.txt

Output:

awhan\_nmbryant\_1.txt

1/2 3/4 == : false

4/6 2/3 == : true

-1/2 1/2 == : false

1 50/50 == : true

1 1/3 == : false

1/2 3/4 != : true

4/6 2/3 != : false

-1/2 1/2 != : true

1 50/50 != : false

1 1/3 != : true

1/2 3/4 < : true

4/6 2/3 < : false

5/4 1/3 < : false

-1/2 1/2 < : true

1 50/50 < : false

1 1/3 < : false

1/3 1 < : true

1/2 3/4 <= : true

4/6 2/3 <= : true

5/4 1/3 <= : false

-1/2 1/2 <= : true

1 50/50 <= : true

1 1/3 <= : false

1/3 1 <= : true

1/2 3/4 > : false

4/6 2/3 > : false

5/4 1/3 > : true

-1/2 1/2 > : false

1 50/50 > : false

1 1/3 > : true

1/3 1 > : false

1/2 3/4 >= : false

4/6 2/3 >= : true

5/4 1/3 >= : true

-1/2 1/2 >= : false

1 50/50 >= : true

1 1/3 >= : true

1/3 1 >= : false

awhan\_nmbryant\_2.txt

7/3 31/47 + : 422/141 (double 2.99291)

36/81 3/9 + : 7/9 (double 0.777778)

5/6 -9/10 + : -1/15 (double -0.0666667)

-1/3 5/-10 + : -5/6 (double -0.833333)

-6/-33 1/2 + : 15/22 (double 0.681818)

3 4/5 + : 19/5 (double 3.8)

-3 7/10 + : -23/10 (double -2.3)

8 -13/2 + : 3/2 (double 1.5)

-7 -8/3 + : -29/3 (double -9.66667)

4 5 + : 9 (double 9)

7/3 31/47 - : 236/141 (double 1.67376)

36/81 3/9 - : 1/9 (double 0.111111)

5/6 -9/10 - : 26/15 (double 1.73333)

-1/3 5/-10 - : 1/6 (double 0.166667)

-6/-33 1/2 - : -7/22 (double -0.318182)

3 4/5 - : 11/5 (double 2.2)

-3 7/10 - : -37/10 (double -3.7)

8 -13/2 - : 29/2 (double 14.5)

-7 -8/3 - : -13/3 (double -4.33333)

4 5 - : -1 (double -1)

7/3 31/47 \* : 217/141 (double 1.53901)

36/81 3/9 \* : 4/27 (double 0.148148)

5/6 -9/10 \* : -3/4 (double -0.75)

-1/3 5/-10 \* : 1/6 (double 0.166667)

-6/-33 1/2 \* : 1/11 (double 0.0909091)

3 4/5 \* : 12/5 (double 2.4)

-3 7/10 \* : -21/10 (double -2.1)

8 -13/2 \* : -52 (double -52)

-7 -8/3 \* : 56/3 (double 18.6667)

4/0 1/2 \* ERROR

4 5 \* : 20 (double 20)

awhan\_nmbryant\_3.txt

7/3 31/47 / : 329/93 (double 3.53763)

36/81 3/9 / : 4/3 (double 1.33333)

5/6 -9/10 / : -25/27 (double -0.925926)

-1/3 5/-10 / : 2/3 (double 0.666667)

-6/-33 1/2 / : 4/11 (double 0.363636)

3 4/5 / : 15/4 (double 3.75)

-3 7/10 / : -30/7 (double -4.28571)

8 -13/2 / : -16/13 (double -1.23077)

-7 -8/3 / : 21/8 (double 2.625)

4/0 1/2 / ERROR

4 5 / : 4/5 (double 0.8)

1/3 9/10 + 2/6 \* 6/5 - : -71/90 (double -0.788889)

5/6 7/8 - 4/7 / : -7/96 (double -0.0729167)