Programming Project #1
EGRE246 Spring 2018
Rational Numbers

1 Overview

Write a C++ class to implement rational numbers as specified below. File `Rat.h` (downloadable off the class web pages):

```
class Rat {
 public:
  Rat();               // constructs 0/1
  Rat(int n_, int d_); // exits prog. with error message if d_==0

  int getN() const;   // getters
  int getD() const;
  void setN(int n_);  // setters
  void setD(int d_);  // exits prog. with error message if d_==0

  Rat norm() const;      // returns normalized number
  Rat reduce() const;    // reduces number to lowest terms
  int cmp(Rat &r) const; // returns -1,0, or 1

  Rat add(Rat& op2) const; // adds 2 numbers
  Rat sub(Rat& op2) const; // subtracts op2 from op1

  friend ostream& operator <<(ostream& os, Rat rat);

 private:
  int n,d;  // numerator, denominator
  int gcd(int a, int b) const; // greatest common divisor
  int lcm(int x, int y); // least common multiple
};
```

Legal rational numbers are to be stored exactly as they are created (i.e. they need not be stored reduced or in normal form). In normal form denominators are always positive and the value 0 has no sign (i.e. 9/-3 normalized would be -9/3, 0/-3 would be 0/3). The arithmetic operators `add` and `sub` always return a number reduced and in normal form. The function `cmp` compares two rational numbers `op1.cmp(op2)` returning -1 if `op1<op2`, 0 if `op1==op2`, and 1 if `op1>op2`.

Here is the implementation for overloaded function `<<`:

```
ostream& operator <<(ostream& os, Rat rat) {
  os << rat.n << '/' << rat.d;
  return os;
}
```

2 Project Documentation

The very first lines in *all* of your programming project files this semester should be, as a minimum, comments giving general information about the project, class, and author using the following format (you of course will *use your name* instead of mine):

```
/* Project #1
   EGRE246 Spring 2018
   D. Resler */
```

In addition, where applicable (i.e. not all projects will require that you turn in an executable main function), the first output for your projects this semester should be the name of the project and your name, e.g.

```
/* ... */
int main() {
  printf("Proj. #1 - D. Resler\n");
  /* ... */
}
```

Again, **be sure to use your name instead of mine!**

3 Sample Driver

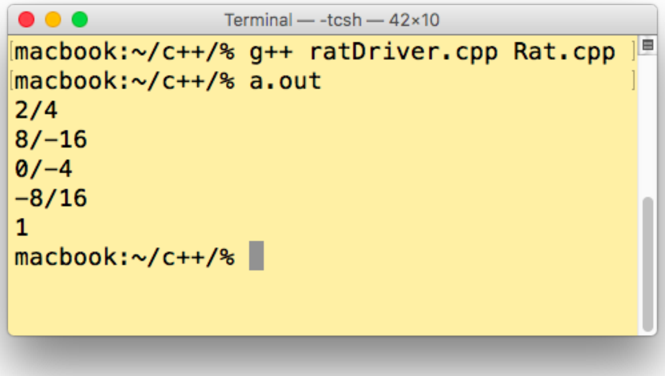
Here is a sample driver program `ratDriver.cpp` that can be used as the first step in testing your class:

```
#include <iostream>
using namespace std;
#include "Rat.h"

int main() {
  Rat r1(2,4), r2(8,-16), r3(0,-4);

  cout << r1 << endl;
  cout << r2 << endl;
  cout << r3 << endl;
  cout << r2.norm() << endl;
  cout << r1.cmp(r2) << endl;

}
```



```
macbook:~/c++/% g++ ratDriver.cpp Rat.cpp
macbook:~/c++/% a.out
2/4
8/-16
0/-4
-8/16
1
macbook:~/c++/%
```

4 Deliverables

You should only turn in your rational class implementation file! I will test your code with my own driver test program. Name your file `Rat`*XXXX*`.cpp` where *XXXX* is the last 4 digits of your student id number. For example, if your student id number is V12345678, your file will be named `Rat5678.cpp` . Projects this term will be submitted via the web using a link off of the class web page (`http://danresler.net/egre246`). Be sure to keep a receipt of your file submission. Note you need not turn in an executable file or your driver program!

Due date: Tuesday, January 30