Source File: `~/2336/02/lab02.(C|CPP|cpp|c++|cc|cxx|cp)`
Input: Under control of `main` function
Output: Under control of `main` function
Value: 2

Create a class called `Rational` for performing various operations with fractions. The specification of the class will be provided. Your task will be to provide the implementation. A main program for testing your implementation will also be provided.

Use a `pair` of `int`s to represent the `private` data of the class—the numerator and the denominator. If the pair is declared as

```
pair<int, int> data;
```

use member `first` to represent the numerator and member `second` the denominator. The implementation should provide two constructors: a default constructor that initializes the numerator to zero and the denominator to one and a second constructor that takes two arguments (the first argument should be stored in the numerator and the second in the denominator). The constructors should **not** store the rational number in reduced form. Additional `public` member functions include:

- "Set" functions for setting the numerator and denominator. The `setDenominator` function should check its argument for validity. If the function receives an argument equal to zero (0), the function should set the denominator to one (1).
- "Get" functions for getting the numerator and denominator.
- Reduction of a `Rational` to lowest terms. Also, `reduce` should modify the denominator of a `Rational` with a zero numerator to be one. Further, a negative `Rational` should ensure that the numerator is negative and the denominator is positive. `Rational`s having both numerator and denominator negative should be modified such that both numerator and denominator are positive.

If u and v are integers, not both zero, we say that their greatest common divisor, $\gcd(u, v)$, is the largest positive integer that evenly divides both u and v. When u and v are both zero, every integer evenly divides zero, so it is convenient to set $\gcd(0, 0) = 0$. When either u or v is zero, define $\gcd(u, 0) = |u|$ and $\gcd(0, v) = |v|$. Provide the implementation of this function as a `private` member of the `Rational` class.

A header file is shown in Figure 1, a sample `main` function for testing your implementation is shown in Figure 2, and a sample execution sequence is shown in Figure 3. To use the `Makefile` as distributed in class, add a target of `lab02` to `targets2srcfiles`.

```
1  #ifndef LAB02_H
2  #define LAB02_H
3
4  #include <utility>
5
6  using namespace std;
7
8  class Rational
9  {
10  public:
11   Rational();                          // default constructor
12   Rational(int num, int denom);        // additional constructor
13   void setNumerator(int num);          // set numerator to num
14   void setDenominator(int denom);      // set denominator to denom
15   int getNumerator() const;            // return numerator
16   int getDenominator() const;          // return denominator
```

Figure 1. `/usr/local/2336/include/lab02.h` (Part 1 of 2)

```
17    void reduce();                        // Reduce to lowest terms and
18                                          //   normalize
19   private:
20    pair<int, int> data;                  // member first represents numerator
21                                          // member second represents denominator
22    int gcd(int m, int n) const;          // returns the gcd of m and n
23  };
24
25  #endif
```

Figure 1. `/usr/local/2336/include/lab02.h` (Part 2 of 2)

```
1   #include <lab02.h>
2   #include <iostream>
3   #include <cstdlib>
4
5   using namespace std;
6
7   int main()
8   {
9     int n, d;
10    Rational first(1, -2), second(-3, 0), third;
11
12    cout << "first = " << first.getNumerator() << '/' << first.getDenominator()
13         << endl;
14    cout << "second = " << second.getNumerator() << '/'
15         << second.getDenominator() << endl;
16    cout << "third = " << third.getNumerator() << '/' << third.getDenominator()
17         << endl;
18
19    while (cin >> n >> d)
20    {
21      third.setNumerator(n);
22      third.setDenominator(d);
23      cout << "Before Reduce() third = "
24           << third.getNumerator() << '/' << third.getDenominator();
25      third.reduce();
26      cout << " After Reduce() third = "
27           << third.getNumerator() << '/' << third.getDenominator() << endl;
28    }
29
30    return EXIT_SUCCESS;
31  }
```

Figure 2. `/usr/local/2336/src/lab02main.C`

```
newuser@csunix ~> cd 2336
newuser@csunix ~/2336> ./getlab.ksh 02
  * Checking to see if a folder exists for Lab 02. . .No
  * Creating a folder for Lab 02
  * Checking to see if Lab 02 has sample input and output files. . .Yes
  * Copying input and output files for Lab 02
    from folder /usr/local/2336/data/02 to folder ./02
  * Checking to see if /usr/local/2336/src/lab02main.C exists. . .Yes
  * Copying file /usr/local/2336/src/lab02main.C to folder ./02
  * Checking to see if /usr/local/2336/include/lab02.h exists. . .Yes
  * Copying file /usr/local/2336/include/lab02.h to folder ./02
  * Copying file /usr/local/2336/src/Makefile to folder ./02
  * Adding a target of lab02 to targets2srcfiles
  * Touching file ./02/lab02.cpp
  * Edit file ./02/lab02.cpp in Notepad++
newuser@csunix ~/2336> cd 02
newuser@csunix ~/2336/02> ls
01.dat       01.out       Makefile     lab02.cpp    lab02.h      lab02main.C
newuser@csunix ~/2336/02> make lab02
g++ -g -Wall -std=c++11 -c lab02main.C -I/usr/local/2336/include -I.
g++ -g -Wall -std=c++11 -c lab02.cpp -I/usr/local/2336/include -I.
g++ -o lab02 lab02main.o lab02.o -L/usr/local/2336/lib -lm -lbits
newuser@csunix ~/2336/02> cat 01.dat
-3  4  3  4
 3 -4 -3 -4
25 45  8 99
 1  0  2  0
129 6579 1935 249
1331 1651 2301 1079
3 1260 6 198
43 1935 207 6579
5 7 -25 -35
-83 1651 127 -1079
1079 1651
newuser@csunix ~/2336/02> cat 01.dat | ./lab02
first = 1/-2
second = -3/1
third = 0/1
Before Reduce() third = -3/4 After Reduce() third = -3/4
Before Reduce() third = 3/4 After Reduce() third = 3/4
Before Reduce() third = 3/-4 After Reduce() third = -3/4
Before Reduce() third = -3/-4 After Reduce() third = 3/4
Before Reduce() third = 25/45 After Reduce() third = 5/9
Before Reduce() third = 8/99 After Reduce() third = 8/99
Before Reduce() third = 1/1 After Reduce() third = 1/1
Before Reduce() third = 2/1 After Reduce() third = 2/1
Before Reduce() third = 129/6579 After Reduce() third = 1/51
Before Reduce() third = 1935/249 After Reduce() third = 645/83
Before Reduce() third = 1331/1651 After Reduce() third = 1331/1651
```

Figure 3. Commands to Compile, Link, & Run Lab 02 (Part 1 of 2)

```
Before Reduce() third = 2301/1079 After Reduce() third = 177/83
Before Reduce() third = 3/1260 After Reduce() third = 1/420
Before Reduce() third = 6/198 After Reduce() third = 1/33
Before Reduce() third = 43/1935 After Reduce() third = 1/45
Before Reduce() third = 207/6579 After Reduce() third = 23/731
Before Reduce() third = 5/7 After Reduce() third = 5/7
Before Reduce() third = -25/-35 After Reduce() third = 5/7
Before Reduce() third = -83/1651 After Reduce() third = -83/1651
Before Reduce() third = 127/-1079 After Reduce() third = -127/1079
Before Reduce() third = 1079/1651 After Reduce() third = 83/127
newuser@csunix ~/2336/02> cat 01.dat | ./lab02 > my.out
newuser@csunix ~/2336/02> diff 01.out my.out
newuser@csunix ~/2336/02>
```

Figure 3. Commands to Compile, Link, & Run Lab 02 (Part 2 of 2)