# rational.py

# An (over) simplified Rational class

class Rational:
    def __init__(self, num, denom = 1):
        def gcd(a, b):
            while b:
                a, b = b, a % b
            return a

        # First normalize sign (always numerator)
        if denom < 0:
            num *= -1
            denom = int(abs(denom))
        # Then reduce
        gcddivisor = gcd(num, denom)
        self.numerator = num / gcddivisor
        self.denominator = denom / gcddivisor

    def add(self, other):
        n = self.numerator * other.denominator + \
            self.denominator * other.numerator
        d = self.denominator * other.denominator
        # gcd will take care of any necessary reductions
        return Rational(n, d)

    # "Overloading" existing operators.
    def __add__(self, other):
        return self.add(other)

    def __sub__(self, other):
        n = self.numerator * other.denominator - \
            self.denominator * other.numerator
        d = self.denominator * other.denominator
        return Rational(n, d)

    def __str__(self):
        if self.denominator == 1:
            return str(self.numerator)
        else:
            return str(self.numerator) + "/" + str(self.denominator)

    def __eq__(self, other):
        return self.numerator * other.denominator == \
            self.denominator * other.numerator

    def __cmp__(self, other):
        diff = self.__sub__(other).numerator
        if diff > 0:
            return 1
        elif diff < 0:
            return -1
        else:
            return 0

    def __lt__(self, other):
        return self.__cmp__(other) < 0

    def __le__(self, other):
        return self.__cmp__(other) <= 0
print "print Rational(-2,-3)"
print Rational(-2,-3)

print "print Rational(-2,3)"
print Rational(-2,3)

print "print Rational(2,-3)"
print Rational(2,-3)

print "print Rational(20,55)"
print Rational(10,55)

print "half = Rational(1,2)"
half = Rational(1,2)

print "quarter = Rational(1,4)"
quarter = Rational(1,4)

print "third = Rational(1,3)"
third = Rational(1,3)

print "sixth = Rational(1,6)"
sixth = Rational(1,6)

print "one = Rational(1)"
one = Rational(1)

print "print str(half + sixth)"
print str(half - sixth)

print "print str(half + sixth)"
print str(half - sixth)

print "str(half + quarter + quarter)"
print str(half + quarter + quarter)

print "half + quarter == Rational(3,4)"
print half + quarter == Rational(3,4)

print "half < quarter"
print half < quarter

print "quarter < third"
print quarter <= third

print "quarter <= (half - quarter)"
print quarter <= (half - quarter)
>>> import rational

>>> print(rational.Rational(-2, 3))
-2/3

>>> print(rational.Rational(-2, 3))
-2/3

>>> print(rational.Rational(2, -3))
-2/3

>>> print(rational.Rational(20, 55))
2/11

>>> half = rational.Rational(1, 2)
>>> quarter = rational.Rational(1, 4)
>>> third = rational.Rational(1, 3)
>>> sixth = rational.Rational(1, 6)
>>> one = rational.Rational(1)

>>> print(str(half + sixth))
1/3

>>> print(str(half + sixth))
1/3

>>> print(str(half + quarter + quarter))
1

>>> half + quarter == rational.Rational(3, 4)
True

>>> half < quarter
False

>>> quarter < third
True

>>> quarter <= (half - quarter)
True

>>>