# NOTE: This is a document that my teacher uploaded online to our school webpage. It’s the code we’re supposed to augment to specific details as mentioned in the description document.

# Rational Numbers Assignment

def gcd(bigger, smaller):

'''compute the greatest common divisor of two positive integers'''

#print(' in gcd ')

if not bigger > smaller :

bigger, smaller = smaller, bigger

while smaller != 0:

remainder = bigger % smaller

#print('gcd calc, big:{}, small:{}, rem:{}'.format(bigger, smaller, remainder))

bigger, smaller = smaller, remainder

return bigger

def lcm(a, b):

'''calculate the least common multiple of two positive integers'''

#print(' in lcm ')

return (a\*b)//gcd(a,b)

class Rational(object):

'''Rational with numerator and denominator. Denominator defaults to 1'''

def \_\_init\_\_(self, numer, denom = 1):

#print('in constructor')

self.numer = numer

self.denom = denom

def \_\_str\_\_(self):

'''String representation for printing'''

#print(' in str ')

return str(self.numer) + '/' + str(self.denom)

def \_\_repr\_\_(self):

''' Used in the interpreter. Call \_\_str\_\_ for now'''

print(' in repr ')

return self.\_\_str\_\_()

def \_\_add\_\_(self, param\_Rational):

'''Add two Rationals'''

if type(param\_Rational) == int:

param\_Rational = Rational(param\_Rational)

if type(param\_Rational) == Rational:

# find the lcm

the\_lcm = lcm(self.denom, param\_Rational.denom)

# multiply each numerator by the lcm, then add

numerator\_sum = the\_lcm\*self.numer/self.denom + \

the\_lcm\*param\_Rational.numer/param\_Rational.denom

return Rational( int(numerator\_sum), the\_lcm )

else:

print("Wrong type in addition method.")

raise(TypeError)

def \_\_sub\_\_(self, param\_Rational):

'''Subtract two Rationals'''

#print(' in add ')

# find the lcm

the\_lcm = lcm(self.denom, param\_Rational.denom)

# multiply each numerator by the lcm, then add

numerator\_sum = the\_lcm\*self.numer/self.denom - \

the\_lcm\*param\_Rational.numer/param\_Rational.denom

return Rational( int(numerator\_sum), the\_lcm )

def reduce\_rational(self):

'''Return the reduced fraction value as a Rational'''

# find the gcd and divide numerator and denominator by it

the\_gcd = gcd(self.numer, self.denom)

return Rational( self.numer//the\_gcd, self.denom//the\_gcd)

def \_\_eq\_\_(self, param\_Rational):

'''Compare two Rationals for equalit and return a Boolean'''

reduced\_self = self.reduce\_rational()

reduced\_param = param\_Rational.reduce\_rational()

return reduced\_self.numer == reduced\_param.numer and\

reduced\_self.denom == reduced\_param.denom

# ------------------------------ MAIN ------------------------------