def main():

pass

if \_\_name\_\_ == '\_\_main\_\_':

main()

class Razlomak(object):

def \_\_init\_\_(self, brojnik, nazivnik):

self.\_brojnik = brojnik

self.\_nazivnik = nazivnik

@property

def brojnik(self):

return self.\_brojnik

@brojnik.setter

def brojnik(self, value):

self.\_brojnik = value

@property

def nazivnik(self):

return self.\_nazivnik

@nazivnik.setter

def nazivnik(self, value):

self.\_nazivnik = value

def skrati(self):

nd = 1

for i in range(1, self.brojnik+1):

if(self.brojnik % i == 0 and self.nazivnik % i == 0):

nd = i

i= i + 1

return self.brojnik / nd, self.nazivnik / nd

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

return "Razlomak("+repr(self.\_brojnik) + ',' + repr(self.\_nazivnik) + ")"

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

return 'Razlomak ' + str(self.\_brojnik) + ' | ' + str(self.\_nazivnik)

def \_\_eq\_\_(self, other):

return self.brojnik/self.nazivnik == other.brojnik/other.nazivnik

def \_\_lt\_\_(self,other):

return self.brojnik/self.nazivnik < other.brojnik/other.nazivnik

def \_\_le\_\_(self,other):

return self.brojnik/self.nazivnik <= other.brojnik/other.nazivnik

def \_\_mul\_\_(self,other):

return Razlomak(self.brojnik \* other.brojnik, self.nazivnik \* other.nazivnik)

def \_\_mod\_\_(self,other):

return Razlomak(self.brojnik \* other.nazivnik, self.nazivnik \* other.brojnik)

def \_\_add\_\_(self,other):

a= self.nazivnik

b= other.nazivnik

nv=1

if(a<b):

nv=a

if(b%a == 0):

b=b/a

else:

b=b

nv=a\*b

else:

nv=b

if(a%b == 0):

a=a/b

else:

a=a

nv=a\*b

return Razlomak(nv / self.nazivnik \* self.brojnik + nv / other.nazivnik \* other.brojnik, nv)

def \_\_sub\_\_(self,other):

a= self.nazivnik

b= other.nazivnik

nv=1

if(a<b):

nv=a

if(b%a == 0):

b=b/a

else:

b=b

nv=a\*b

else:

nv=b

if(a%b == 0):

a=a/b

else:

a=a

nv=a\*b

return Razlomak (nv / self.nazivnik \* self.brojnik - nv / other.nazivnik \* other.brojnik, nv)