

Class_Rational_1

คลาส **rational** ข้างล่างนี้แทน **จำนวนตรรกยะ** ที่รองรับการทำงานของคำสั่ง **str** กับ **float** และเครื่องหมาย < กับ *

คลาส rational	ตัวอย่างการใช้งานคลาส rational
<pre>class rational: def __init__(self, n, d): self.n = n self.d = d def __str__(self): return str(self.n) + "/" + str(self.d) def __lt__(self, rhs): return float(self) < float(rhs) def __float__(self): return self.n / self.d def __mul__(self, rhs): return rational(self.n*rhs.n, self.d*rhs.d)</pre>	<pre>a = rational(2,4) b = rational(3,9) print(str(a), str(b)) # 2/4 3/9 print(b < a) # True a = a * a print(str(a)) # 4/16 print(a) # 4/16 print(float(a)) # 0.25</pre>

เมทอด **__lt__** ถูกเรียกเมื่อเราใช้ตัวปฏิบัติการ < กับ **rational** สองตัว เพื่อเปรียบเทียบว่าตัวซ้ายน้อยกว่าตัวขวาหรือไม่

เมทอด **__str__** ถูกเรียกเมื่อคำสั่ง **str(a)** ทำงาน โดยที่ **a** เป็น **rational** ได้ผลลัพธ์เป็นสตริงที่แทนค่าของ **a**

เมทอด **__float__** ถูกเรียกเมื่อคำสั่ง **float(a)** ทำงาน โดยที่ **a** เป็น **rational** ได้ผลลัพธ์เป็น **float** ที่แทนค่าของ **a**

เมทอด **__mul__** ถูกเรียกเมื่อเราใช้ตัวปฏิบัติการ * กับ **rational** สองตัว ได้ผลลัพธ์เป็น **rational** ใหม่ที่แทนผลคูณที่ได้

จงปรับปรุง

จงปรับปรุงคลาส **rational** ให้รองรับการ + - และ / จำนวนตรรกยะด้วย โดยเพิ่มเมทอด **__add__** **__sub__** และ **__truediv__** ตามลำดับ

การส่งตรวจ

ให้นำโปรแกรมข้างล่างนี้ ต่อท้าย **class rational** ที่เขียนข้างบนนี้ แล้วจึงส่งให้ grader ตรวจ

```
t, n1, d1, n2, d2 = input().split()
r1 = rational(int(n1),int(d1)); r2 = rational(int(n2),int(d2))
if t == '+' : print(float(r1 + r2))
elif t == '-' : print(float(r1 - r2))
else :      print(float(r1 / r2))
```

ข้อมูลนำเข้า

สตริง 5 ตัว คั่นด้วยช่องว่าง (ดูตัวอย่าง และโปรแกรมที่ส่งตรวจประกอบ)

ข้อมูลส่งออก

ผลการทำงานของโปรแกรมข้างบนที่อาศัยคลาส **rational** ที่เขียน

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
+ 2 4 3 4	1.25
- 2 4 3 4	-0.25
/ 2 4 3 4	0.6666666666666666