

Class_Reverse_int

จำนวนเต็มที่เราเรียนมาเป็นแบบที่เรียกว่า **int** โจทย์ข้อนี้จะมาสร้างประเภทข้อมูลใหม่เรียกว่า **rint** เป็นจำนวนเต็มที่เขียนกลับลำดับที่เราเขียนกัน เช่น ถ้าเขียน `a = rint("0021")` `a` คือ 1200 แบบ **int** โจทย์ข้อนี้ให้เขียนคลาส **rint** ที่มีโครงและตัวอย่างการใช้งานดังนี้

โครงของคลาส rint	ตัวอย่างการใช้งาน rint
<pre>class rint : def __init__(self, r): def __lt__(self, rhs): def __str__(self): def __int__(self): def __add__(self, rhs):</pre>	<pre>a = rint("08100") b = rint("91") print(a < b) # ได้ False print(str(a)) # ได้ 081 print(int(a)) # ได้ 180 c = a + b print(str(c)) # ได้ 991</pre>

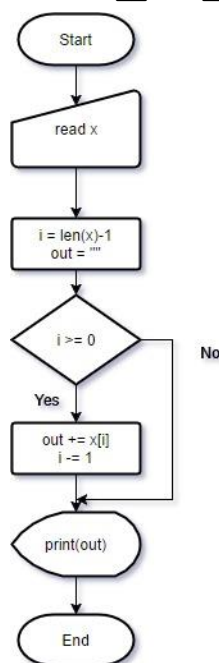
เมทอด `__lt__` ถูกเรียกเมื่อเราใช้ตัวปฏิบัติการ `<` กับ **rint** สองตัว เพื่อเปรียบเทียบว่าตัวซ้ายน้อยกว่าตัวขวาหรือไม่

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

เมทอด `__int__` ถูกเรียกเมื่อคำสั่ง `int(a)` ทำงาน โดยที่ `a` เป็น **rint** ได้ผลลัพธ์เป็น **int** ที่แทนค่าของ `a`

เมทอด `__add__` ถูกเรียกเมื่อเราใช้ตัวปฏิบัติการ `+` กับ **rint** สองตัว ได้ผลลัพธ์เป็น **rint** ใหม่ที่แทนผลบวกที่ได้

ให้นักศึกษาใช้ flowchart ด้านล่างเพื่อเป็นแนวทางสำหรับการเขียนเมทอด `__init__` ของ class **rint**



การส่งตรวจ

ให้นำโปรแกรมข้างล่างนี้ ต่อท้าย `class rint` ที่เขียนข้างบนนี้ แล้วจึงส่งให้ grader ตรวจ

```
t, r1, r2 = input().split()
a = rint(r1); b = rint(r2)
if t == '1' : print(a < b)
elif t == '2' : print(int(a),int(b))
elif t == '3' : print(str(a),str(b))
elif t == '4' : print(int(a + b))
else : print(str(a + b))
```

ข้อมูลนำเข้า

สตริง 3 ตัว คั่นด้วยช่องว่าง (ดูตัวอย่าง และโปรแกรมที่ส่งตรวจประกอบ)

ข้อมูลส่งออก

ผลการทำงานของโปรแกรมข้างบนที่อาศัยคลาส `rint` ที่เขียน

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
1 10 01	True
1 0123 1230	False
2 0123 1230	3210 321
3 0123 123	0123 123
4 0123 1230	3531
5 0123 1230	1353