

All Paths

กำหนดให้ มีเมืองหลายเมือง แต่ละเมืองแทนด้วยเลขจำนวนเต็ม สมมติว่า เรามีทางเดินที่เป็นลำดับของเลขเมืองดังนี้

3 5 8
3 5 6 7

เราสามารถเก็บทางเดินเหล่านี้ด้วย `list` เป็น `[[3,5,8], [3,5,6,7]]`

จะเห็นมีทางเดินส่วนหน้าซ้ำกัน สามารถเก็บแบบทางเดินร่วมแยกออกมาเป็น `[3,[5,[8],[6,[7]]]]`

ซึ่งขออธิบายแนวทางเก็บแบบหลังนี้ด้วยตัวอย่างในตารางข้างล่างนี้

<code>list</code> ที่เก็บทางเดินแบบ A	<code>list</code> ที่เก็บทางเดินแบบ B
<code>[[3]]</code>	<code>[3]</code>
<code>[[3,4]]</code>	<code>[3,[4]]</code>
<code>[[3,4,5]]</code>	<code>[3,[4,[5]]]</code>
<code>[[7], [8,9]]</code>	<code>[[7],[8,[9]]]</code>
<code>[[3,7], [3,8,9]]</code>	<code>[3,[[7],[8,[9]]]]</code>
<code>[[3,7], [3,8,9], [2,4,5]]</code>	<code>[[3,[[7],[8,[9]]]], [2,[4,[5]]]]</code>
<code>[[1,3,7], [1,3,8,9], [1,2,4,5]]</code>	<code>[1, [[3,[[7],[8,[9]]]], [2,[4,[5]]]]]</code>

จงเขียนฟังก์ชัน `B_to_A` ที่รับ `list` ทางเดินแบบ **B** แล้วแปลงให้เป็น `list` ทางเดินแบบ **A**

```
def B_to_A( b ) :  
  
    ???  
  
print(B_to_A(eval(input().strip()))) # do not remove this line
```

ข้อมูลนำเข้า

list แบบ B

ข้อมูลส่งออก

list แบบ A

ตัวอย่าง

input	Output
<code>[3]</code>	<code>[[3]]</code>
<code>[3,[4]]</code>	<code>[[3, 4]]</code>
<code>[3,[4,[5]]]</code>	<code>[[3, 4, 5]]</code>
<code>[[7], [8,[9]]]</code>	<code>[[7], [8, 9]]</code>
<code>[3,[[7],[8,[9]]]]</code>	<code>[[3, 7], [3, 8, 9]]</code>
<code>[[3,[[7],[8,[9]]]], [2,[4,[5]]]]</code>	<code>[[3, 7], [3, 8, 9], [2, 4, 5]]</code>
<code>[1, [[3,[[7],[8,[9]]]], [2,[4,[5]]]]]</code>	<code>[[1, 3, 7], [1, 3, 8, 9], [1, 2, 4, 5]]</code>