

Matrix Sum

เมทริกซ์ของจำนวนเต็มขนาด $r \times c$ สามารถแทนได้ด้วยลิสต์ขนาด r ช่องโดยที่แต่ละช่องเก็บลิสต์ขนาด c ช่อง เช่น

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$
 แทนได้ด้วย `[[1, 2, 3], [4, 5, 6]]`

จงเขียนฟังก์ชัน `m_sum(m, axis)` ที่คืนลิสต์ของผลรวมของจำนวนในเมทริกซ์ `m` ตามแนวแกน `axis` ที่กำหนด

ถ้า `axis=0` ให้หาผลรวมของทุก ๆ แถวในแต่ละคอลัมน์

ถ้า `axis=1` ให้หาผลรวมของทุก ๆ คอลัมน์ในแต่ละแถว

เช่น ให้ `m` เป็นเมทริกซ์ ดังตัวอย่างด้านบน

`m_sum(m, 0)` จะได้ `[5, 7, 9]`

`m_sum(m, 1)` จะได้ `[6, 15]`

```
def m_sum(m, axis):
```

```
exec(input().strip()) #ต้องมีบรรทัดนี้เมื่อส่งไป grader
```

*** ห้าม import อะไรเพิ่ม *** (Grader ไม่ได้ตรวจเรื่องนี้ แต่จะตรวจในภายหลัง)

ข้อมูลนำเข้า

คำสั่งในการทดสอบฟังก์ชันที่เขียน

ข้อมูลส่งออก

ผลที่ได้จากคำสั่งที่ป้อนเป็นข้อมูลนำเข้า

ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
<code>print(m_sum([[1,2],[3,4]],0))</code>	<code>[4, 6]</code>
<code>print(m_sum([[1,2],[3,4]],1))</code>	<code>[3, 7]</code>
<code>print(m_sum([[0,0],[0,0],[0,0]],1))</code>	<code>[0, 0, 0]</code>
<code>A=[[1,2],[3,4],[5,6]];print(m_sum(A,0), m_sum(A,1))</code>	<code>[9, 12] [3, 7, 11]</code>
<code>A=[[17]];print(m_sum(A,0), m_sum(A,1))</code>	<code>[17] [17]</code>