

Summer 2023: Euro 2024 Sub-routing

หนังสือพิมพ์ฉบับหนึ่งเชิญชวนให้อ่านเข้าร่วมกิจกรรมทายผลการแข่งขันฟุตบอลยูโร 2024 ซึ่งต้องมีการจัดข้อมูลให้สามารถประมวลผลได้ง่าย ดังนั้นโจทย์ข้อนี้จึงต้องการให้นิสิตเขียนฟังก์ชัน `extract_scores()`

```
def extract_scores(actual_results, guess_results):  
    # TODO: Write your code here!  
  
# TODO: DO NOT modify the code below this line  
exec(input().strip())  
exec(input().strip())  
exec(input().strip())
```

ซึ่งฟังก์ชัน `extract_scores()` มีรายละเอียดดังนี้

- `extract_scores()` จะรับ input เป็น list ของข้อความทั้งหมด 2 list และจะคืนค่าเป็น list ที่ประกอบด้วย list ย่อย ที่เก็บผลจริง และการทายผลของแต่ละคู่
- `actual_results` คือ list ที่เก็บผลการแข่งขันจริงทั้งหมด
- `guess_results` คือ list ที่เก็บการทายผลของผู้เข้าร่วมกิจกรรม

ตัวอย่าง `actual_results` และ `guess_results`

```
actual_results = [  
    "England:Germany 2:0",  
    "France:Spain 1:1",  
    "Portugal:Italy 0:3",  
    "Netherlands:Belgium 2:1"  
]  
guess_results = [  
    "Germany:England 2:1",  
    "Portugal:Italy 1:1",  
    "France:Spain 2:0"  
]
```

ถ้าหากมีการเรียกใช้ฟังก์ชัน `extract_scores()` โดยรับลิสต์ตัวอย่างข้างบน ดังคำสั่ง

```
extract_scores(actual_results, guess_results)
```

ภายในฟังก์ชันจะพิจารณาผลคะแนนในการแข่งขันต่างๆ เรียงลำดับตาม `actual_results` ดังตาราง

การแข่งขัน	ผลจริง	ผลทาย	ลิสต์ย่อยของแต่ละคู่การแข่งขัน
England : Germany	2 : 0	1 : 2	[2, 0, 1, 2]
France : Spain	1 : 1	2 : 0	[1, 1, 2, 0]
Portugal : Italy	0 : 3	1 : 1	[0, 3, 1, 1]
Netherlands : Belgium	2 : 1	-	[]

เมื่อประมวลผลเสร็จแล้ว ฟังก์ชัน จะคืนค่าเป็น list ซ้อน list ที่มีลักษณะดังนี้

```
[[2, 0, 1, 2], [1, 1, 2, 0], [0, 3, 1, 1], []]
```

หมายเหตุ: หากคู่ไหนไม่มีการทายผล ให้คืนค่า list ย่อย ออกมาเป็น list ว่าง ([])

Input (ข้อมูลนำเข้า)

ข้อมูลนำเข้า จะมีทั้งหมด 3 บรรทัด มีลักษณะเป็นคำสั่งในภาษา Python

- บรรทัดที่ 1 จะกำหนดค่าของ list `actual_results`
- บรรทัดที่ 2 จะกำหนดค่าของ list `guess_results`
- บรรทัดที่ 3 จะเป็นคำสั่งแสดงผลจากการใช้งานฟังก์ชัน `extract_scores()`

เงื่อนไขของข้อมูลนำเข้า

- รับประกันว่าทุกคู่ใน `guess_results` จะพบใน `actual_results` เสมอ
แต่อาจมีการสลับลำดับ เช่น สลับจาก “England:Germany” เป็น “Germany:England”
- ผู้เข้าร่วมกิจกรรม ไม่จำเป็นต้องทายผลครบทุกคู่

Output (ผลลัพธ์)

แสดงผลที่ได้จากการคืนค่าของฟังก์ชัน `extract_scores()` ซึ่งจะมีลักษณะเป็น list ที่ประกอบไปด้วย list ย่อยตามจำนวนคู่ใน `actual_results` และภายใน list ย่อย จะประกอบไปด้วยผลจริง ตามด้วยผลทาย เรียงลำดับทีม และลำดับคู่การแข่งขัน ตามลิสต์ หากไม่มีการทายผลในคู่ นั้น ให้ใส่เป็น list ว่างแทน ([])

Examples (ตัวอย่าง)

ตัวอย่างที่ 1

Input
<pre>actual_results = ["England:Germany 2:0", "France:Spain 1:1", "Portugal:Italy 0:3", "Netherlands:Belgium 2:1"] guess_results = ["England:Germany 2:0", "France:Spain 1:1", "Portugal:Italy 0:3", "Netherlands:Belgium 2:1"] print(extract_scores(actual_results, guess_results))</pre>
Output
<pre>[[2, 0, 2, 0], [1, 1, 1, 1], [0, 3, 0, 3], [2, 1, 2, 1]]</pre>

ตัวอย่างที่ 2

Input
<pre>actual_results = ["England:Germany 2:0", "France:Spain 1:1", "Portugal:Italy 0:3", "Netherlands:Belgium 2:1"] guess_results = ["Germany:England 1:2", "Spain:France 2:1", "Italy:Portugal 3:2", "Belgium:Netherlands 2:1"] print(extract_scores(actual_results, guess_results))</pre>
Output
<pre>[[2, 0, 2, 1], [1, 1, 1, 2], [0, 3, 2, 3], [2, 1, 1, 2]]</pre>

ตัวอย่างที่ 3

Input
<pre>actual_results = ["England:Germany 2:0", "France:Spain 1:1", "Portugal:Italy 0:3", "Netherlands:Belgium 2:1"] guess_results = ["Germany:England 2:1", "Portugal:Italy 1:1", "France:Spain 2:0"] print(extract_scores(actual_results, guess_results))</pre>
Output
<pre>[[2, 0, 1, 2], [1, 1, 2, 0], [0, 3, 1, 1], []]</pre>

คำแนะนำ

เพื่อให้การแก้ไขง่ายขึ้น นิสิตสามารถเขียนฟังก์ชัน `get_score()` ให้คืนค่าผลคะแนนของการแข่งขันระหว่าง `team01` และ `team02` บน list `results` ได้

```
def get_score(team01, team02, results):  
    # TODO: This returns the score of a match between  
    #         team01 and team02 in results.
```

หมายเหตุ: ชื่อทีมบน `results` สามารถสลับกันได้