

FRA 141 การเขียนโปรแกรมคอมพิวเตอร์สำหรับวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ 1

(Computer Programming for Robotics and Automation Engineering I)

ภาคเรียนที่ 1 ปีการศึกษา 2565

สถาบันวิทยาการหุ่นยนต์ภาคสนาม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

หัวข้อ: การแก้ปัญหาด้วยการเขียนโปรแกรมภาษาไพธอน

ส่งงานที่: submit.fibo.kmutt.ac.th

วิธีการสอบ

เขียนฟังก์ชัน ต่อไปนี้ในภาษาไพธอน

* เขียนชื่อฟังก์ชันให้ถูกต้อง ตัวพิมพ์เล็กพิมพ์ใหญ่ต้องให้เหมือนกับโจทย์ ถ้าชื่อฟังก์ชันผิด แล้วทดสอบไม่ผ่านจะเสียคะแนน

* โจทย์จะกำหนดว่าให้ฟังก์ชัน return ผลลัพธ์อะไรออกมา ใช้คำสั่ง return ตามโจทย์ อย่าใช้วิธี print ผลลัพธ์ออกมา เพราะจะทดสอบไม่ผ่านและเสียคะแนน

* โจทย์แต่ละข้อจะมีตัวอย่างให้ โดยจะประกอบไปด้วย คำสั่งให้ลองเรียกและผลลัพธ์ที่ควรจะได้ นักศึกษาควรพยายามแก้ไขจนกว่าจะได้ผลลัพธ์ที่ถูกต้องทุกอัน เพื่อให้แน่ใจว่าได้คะแนนเต็ม

* คะแนนเต็ม 10 คะแนน

Gold#05: Let's Jump

เกมส์กระโดดแสนสนุกมีกฎดังต่อไปนี้ เรามีเลนอยู่สองเลน แต่ละเลนจะมีแท่น (เครื่องหมาย '.') กับมีท่อ (เครื่องหมาย '#') เช่น

'..##..'

'..#...'

เลนแรกมีแท่นในตำแหน่งที่ 0,1,4,5 และท่อที่ตำแหน่ง 2,3

เลนที่สองมีแท่นในตำแหน่งที่ 0,1,3,4,5 และท่อที่ตำแหน่ง 2

นักกระโดดของเราจะเริ่มที่แท่นแรกบนเลนแรกเสมอ และจะกระโดดจากซ้ายไปขวาไปจนถึงแท่นสุดท้ายของเลนใดก็ได้ก็จะเรียกว่าถึงเส้นชัย

ในการกระโดดแต่ละครั้ง นักกระโดดจะทำได้สามอย่าง

1. ถ้าเขาอยู่บนเลน L ตำแหน่งที่ X เขาสามารถกระโดดไปยังแท่นในตำแหน่งที่ X+1 บนเลนเดียวกันได้
2. ถ้าเขาอยู่บนเลน L ตำแหน่งที่ X เขาสามารถกระโดดไปยังแท่นในตำแหน่งที่ X บนอีกเลนหนึ่งได้
3. ถ้ามีท่อขวางอยู่หนึ่งอันบนเลนที่เขาอยู่เขาจะสามารถกระโดดข้ามมันได้ แต่ถ้ามีท่อขวางอยู่สองอันเขาจะกระโดดข้ามไม่ได้แล้ว

เขียนฟังก์ชัน two_lanes ที่รับค่าเป็นเลนทั้งสองเลนแล้วรีเทิร์นค่าเป็นจำนวนการกระโดดทั้งหมดที่น้อยที่สุดที่นักกีฬาต้องกระโดดเพื่อไปถึงเส้นชัย

ถ้ามันเป็นไปไม่ได้ที่เขาจะสามารถกระโดดไปยังเส้นชัย รีเทิร์นค่า -1

ตัวอย่าง

two_lanes(['.#.....', '..##....'])

```
>> 5
```

```
two_lanes(['##..', '#...'])
```

```
>> 4
```

```
two_lanes(['##..', '##..'])
```

```
>> -1
```

Note: สองตำแหน่งสุดท้ายจะไม่มีท่อนทั้งสองเลน