

# Programming Fundamentals I Lab.

## 6. ทักษะการแก้ปัญหา

ชื่อ \_\_\_\_\_ รหัสนิต \_\_\_\_\_

ในปฏิบัติการนี้ คุณจะได้อีกการประยุกต์ใช้เครื่องมือและคำสั่งต่าง ๆ ในการเขียนโปรแกรมภาษา Python เพื่อแก้ปัญหา

### 1. Frog

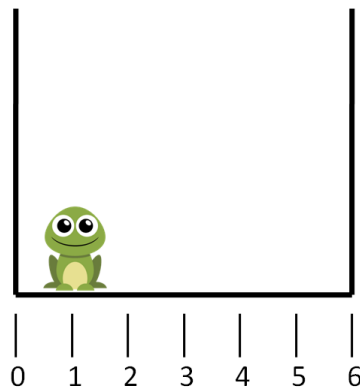
กบตัวหนึ่งถูกขังอยู่ในถังไม้ความกว้าง  $w$  เมื่อเริ่มต้น กบตัวนี้อยู่ในตำแหน่งที่  $s$  จากด้านซ้าย (ขอบซ้ายนับเป็นตำแหน่งที่ 0) จากนั้นกบตัวนี้จะเริ่มกระโดดไปทางขวาด้วยระยะทางครึ่งละ 1 หากระยะทางขวาในถังเหลือน้อยกว่า 1 กบจะกระโดดไปชนขอบถึงและด้งกลับมาจนระยะทางรวมเท่ากับ 1 และด้วยความมั่นใจที่ชนขอบถึง ทำให้ในการกระโดดหลังจากนั้นมันจะกระโดดไปด้านซ้ายแทนจนกว่าจะไปชนขอบถึงด้านซ้าย เป็นเช่นนี้ไปเรื่อย ๆ หากกบกระโดดเป็นจำนวน  $n$  ครั้ง ให้คุณเขียนฟังก์ชันเพื่อหาตำแหน่งหลังการกระโดดแต่ละครั้ง

จงเขียนฟังก์ชันชื่อ `frog()` ที่รับพารามิเตอร์ 4 ตัวได้แก่ ความกว้างของถัง ตำแหน่งเริ่มต้น ระยะกระโดด และจำนวนครั้งที่กระโดด และให้ `return` เป็น list ของตำแหน่งก่อนเริ่มกระโดด ต่อด้วยตำแหน่งหลังกระโดดแต่ละครั้งตามลำดับ

ตัวอย่างการใช้งาน

```
>>> frog(6, 1, 3, 10)
[1, 4, 5, 2, 1, 4, 5, 2, 1, 4, 5]
```

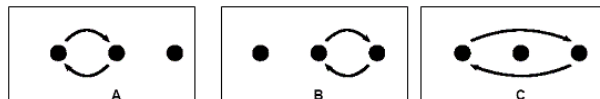
อธิบายตัวอย่าง



จากภาพ เมื่อเริ่มต้นกบอยู่ตำแหน่งที่ 1 กระโดดครั้งแรกกระโดด 3 ไปอยู่ตำแหน่งที่ 4 กระโดดครั้งที่ 2 กระโดดไปชนขอบทางด้านขวาและด้กลับมามาอยู่ตำแหน่งที่ 5 กระโดดครั้งที่ 3 กระโดดไปอยู่ตำแหน่งที่ 2 เป็นเช่นนี้ไปเรื่อย ๆ เมื่อกระโดดครบ 10 ครั้ง จะไปอยู่ที่ตำแหน่งที่ 5

## 2. Trik

สมชายไปเที่ยวงานวัดแห่งหนึ่งซึ่งมีการแสดงมายากลในการสลับลูกบอลที่อยู่ในแก้วทึบ นักมายากลมีแก้วอยู่สามใบวางเรียงกัน โดยลูกบอลจะใส่ไว้ในแก้วใบซ้ายมือ นักมายากลจะสลับแก้วทีละสองใบด้วยความรวดเร็วแต่เราก็คงสามารถมองเห็นได้ว่าเขาจับคู่ไหนหมุนอยู่บ้าง บังเอิญว่าสมชายเป็นโปรแกรมเมอร์จึงเขียนโปรแกรมเพื่อช่วยคำนวณตำแหน่งของลูกบอล โดยที่ได้ทำการเข้ารหัสการสลับของนักมายากลเป็นข้อมูล A B C ดังรูป



จงเขียนฟังก์ชันชื่อ `trik()` ที่รับรหัสการเคลื่อนย้ายแก้วเป็นสตริง และคืนค่าเป็นตำแหน่งสุดท้ายของลูกบอล โดยจะเป็น 1 ถ้าลูกบอลอยู่แก้วซ้ายสุด, 2 ถ้าลูกบอลอยู่แก้วใบกลาง และ 3 ถ้าลูกบอลอยู่ในแก้วขวาสุด

ตัวอย่างการใช้งาน

```
>>> trik('AB')
3
>>> trik('CBABCACCC')
1
```

ที่มา: Croatian Open Competition in Informatics

## 3. Max Sequence

กำหนดให้  $a_1, a_2, \dots, a_n$  เป็น ลำดับ ของจำนวนเต็ม และ กำหนดให้  $a_i, a_{i+1}, \dots, a_j$  เป็น ลำดับย่อย ของ ลำดับดังกล่าวนี้ โดยที่  $i$  และ  $j$  เป็นจำนวนเต็มบวก และ  $1 \leq i \leq j \leq n$  หรือกล่าวอีกนัยหนึ่งคือสมาชิกทุกตัวของ ลำดับย่อย ต้องมีตำแหน่งต่อเนื่องกัน ลำดับย่อย อาจมีได้หลายชุด เมื่อหาผลบวกของสมาชิกทุกตัวใน ลำดับย่อย แต่ละชุด ผลบวกที่ได้ อาจมีค่าแตกต่างกัน

ลำดับย่อย ที่มีผลบวกของสมาชิกสูงสุด เรียกว่า ลำดับย่อยที่มีค่าสูงสุด ซึ่งอาจมีเพียงชุดเดียวหรืออาจมีหลายชุดก็ได้ ในกรณีที่ ลำดับย่อยที่มีค่าสูงสุด มีค่าน้อยกว่าหรือเท่ากับศูนย์ เรียกว่า ลำดับย่อยว่าง (Empty sequence)

ตัวอย่าง ลำดับ  $[4, -6, 3, -2, 6, -4, -6, 6]$  มี ลำดับย่อยที่มีค่าสูงสุด เพียงชุดเดียว คือ ลำดับย่อย  $[3, -2, 6]$  โดยผลบวกของ ลำดับย่อย มีค่าเท่ากับ 7 ลำดับ  $[-2, -3, -1]$  ไม่มี ลำดับย่อย ใดที่มีผลบวกมากกว่าศูนย์ ถือว่ามี ลำดับย่อยว่าง

จงเขียนฟังก์ชันชื่อ `max_sequence()` เพื่อรับ list ที่ประกอบด้วยค่าสมาชิกทุกตัวของลำดับ ทำการคำนวณและคืนค่าเป็นลำดับย่อยที่มีค่าสูงสุด ถ้าลำดับย่อยที่มีค่าสูงสุดเป็น Empty sequence ให้คืนค่าเป็น list ว่าง

ตัวอย่างการใช้งาน

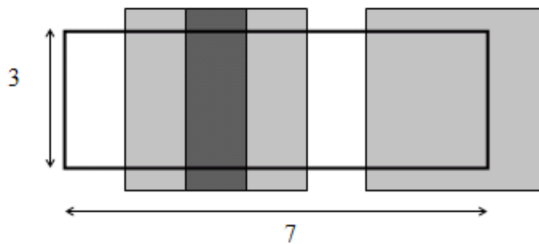
```
>>> max_sequence([4, -6, 3, -2, 6, -4, -6, 6])
[3, -2, 6]
>>> max_sequence([-2, -3, -1])
[]
```

ที่มา: การแข่งขันคอมพิวเตอร์โอลิมปิก สอวน. ครั้งที่ 2 มหาวิทยาลัยบูรพา

#### 4. Filter

หอประชุมแห่งหนึ่งมีหน้าต่างขนาดใหญ่รูปสี่เหลี่ยมผืนผ้า ขนาดกว้าง  $W$  เมตร สูง  $H$  เมตร เนื่องจากฤดูนี้เป็นฤดูร้อน นักศึกษาจึงพยายามลดความร้อนโดยการซื้อผ้าม่านกรองแสงมา  $n$  ผืนและนำมาแขวนที่ตำแหน่งต่าง ๆ ในแนวดิ่งเพื่อบังแดด ผ้าม่านที่ซื้อมามีความกว้างแตกต่างกัน แต่ทุกผืนมีความสูงมากกว่าความสูงของหน้าต่าง (สูงกว่า  $H$  เมตร) ผ้าม่านแต่ละผืนมีความสามารถในการตัดแสงแดดได้ 50% และหากผ้าม่านซ้อนกันมากกว่าหรือเท่ากับสองชั้นสามารถบังแดดได้ 100% ผ้าม่านสามารถแขวนซ้อนกันบางส่วนหรือทั้งหมดก็ได้ และสามารถแขวนซ้อนกันได้มากกว่าหนึ่งผืน

ตัวอย่างของการแขวนผ้าม่านและการบังแดดแสดงดังรูปด้านล่าง ที่มีหน้าต่างกว้าง 7 เมตร สูง 3 เมตร และมีผ้าม่าน 3 ผืน โดยมีสองผืนซ้อนทับกันอยู่



จงเขียนฟังก์ชันชื่อ `filter()` ที่รับข้อมูลขนาดของหน้าต่างและการแขวนผ้าม่าน จากนั้นคำนวณหาพื้นที่ของหน้าต่างที่ไม่โดนม่านบัง (แสงผ่านได้ 100%) และพื้นที่ที่แสงสามารถส่องผ่านได้ 50% มีหน่วยเป็นตารางเมตร

พารามิเตอร์ของฟังก์ชันได้แก่  $W$ ,  $H$ , และ list ของข้อมูลผ้าม่านผืนที่ 1 ถึง  $n$  ตามลำดับ โดยข้อมูลของผ้าม่านแต่ละผืนจะเป็น tuple  $(x, a)$  โดย  $x$  แทนตำแหน่งนับจากขอบหน้าต่างด้านซ้ายที่เริ่มแขวนผ้าม่าน และ  $a$  แทนความกว้างของผ้าม่าน มีหน่วยเป็นเมตร ผ้าม่านจะบังแดดจากหน้าต่างเริ่มจากขอบด้านซ้าย  $x$  เมตรถึง  $x + a$  เมตร

ฟังก์ชันจะต้องคืนค่าเป็น tuple  $(u, v)$  โดย  $u$  เป็นพื้นที่ของหน้าต่างที่แสงส่องผ่านได้โดยไม่โดนม่านบัง (แสงผ่านได้ 100%) และ  $v$  เป็นพื้นที่ของหน้าต่างที่แสงส่องผ่านได้ 50%

ตัวอย่างการใช้งาน

```
>>> filter(7, 3, [(1, 2), (5, 3), (2, 2)])
(6, 12)
```

ที่มา: การแข่งขันคอมพิวเตอร์โอลิมปิก สอวน. ครั้งที่ 3 มหาวิทยาลัยขอนแก่น

## 5. Nugget Number

ร้านฟาสต์ฟู้ดแห่งหนึ่งขายนักเก็ตเป็นกล่อง มีกล่องนักเก็ตอยู่สามขนาด คือ เล็ก, กลาง, และใหญ่ ใส่นักเก็ตจำนวน 6, 9, และ 20 ชิ้นตามลำดับ

เลขนักเก็ต คือจำนวนเต็มบวกที่เกิดจากผลรวมของจำนวนนักเก็ตในกล่องขนาดต่าง ๆ เช่น เลข 6 เป็นเลขนักเก็ต เพราะเป็นจำนวนนักเก็ตในกล่องเล็ก, เลข 12 เป็นเลขนักเก็ตเพราะเกิดจากการรวมกันของจำนวนนักเก็ตในกล่องเล็กสองกล่อง, เลข 15 เป็นเลขนักเก็ตเพราะเกิดจากการรวมกันของจำนวนนักเก็ตในกล่องเล็กหนึ่งกล่องและกล่องกลางหนึ่งกล่อง เป็นต้น เลข 4 และ 10 ไม่เป็นเลขนักเก็ตเพราะเลขดังกล่าวไม่สามารถเกิดจากการรวมกันของจำนวนนักเก็ตในกล่องขนาดใด ๆ ได้

จงเขียนฟังก์ชัน `nugget_numbers()` ที่รับจำนวนเต็มบวก `n` และคืนค่าเป็น list ของเลขนักเก็ตทั้งหมดที่มีค่าน้อยกว่าหรือเท่ากับค่า `n` เรียงจากน้อยไปมาก

ตัวอย่างการใช้งาน

```
>>> nugget_numbers(15)
[6, 9, 12, 15]
>>> nugget_numbers(4)
[]
```

ที่มา: การแข่งขันคอมพิวเตอร์โอลิมปิก สอวน. ครั้งที่ 1 มหาวิทยาลัยเกษตรศาสตร์