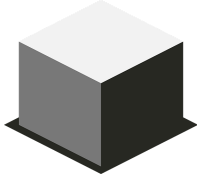




CodeCube The Mini Contest #3

Editorial



## 1. Rock Paper Scissors by PalmPTSJ

---

สำหรับข้อนี้ เราจะมีโอกาสชนะมากที่สุด เมื่อเราเลือกออกของ ที่จะชนะของอีกฝ่าย  
มีโอกาสออกมากที่สุด เช่น อีกฝ่ายมีโอกาสออกหินมากที่สุด เราก็ควรจะออกกระดาษ เพื่อให้โอกาสชนะ  
ของเรามากที่สุด แต่ในข้อนี้จะมีกรณีที่โอกาสรวมไม่เท่ากับ 100 และมีกรณีที่โอกาสที่อีกฝ่ายจะออกของ 2  
ขั้นที่มีโอกาสออกเท่ากัน และมากที่สุด ดังนั้นต้องทำการเช็คเพิ่มเติมด้วย ซึ่งจะเขียนเป็นลำดับการทำงานได้ดังนี้

input a, b, c

if  $a+b+c \neq 100$  :

output "BUG"

end of program // จบการทำงาน เพราะถ้ารวมกันไม่ได้ 100 ก็ไม่แสดงอะไรต่อแล้ว

if  $a > b$  and  $a > c$  :

output "PAPER" // a มากที่สุด (หิน) เราต้องออกกระดาษ

else if  $b > a$  และ  $b > c$  :

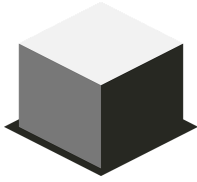
output "SCISSORS" // b มากที่สุด (กระดาษ) เราต้องออกกรรไกร

else if  $c > a$  and  $c > b$  :

output "ROCK" // c มากที่สุด (กรรไกร) เราต้องออกหิน

else :

output "I DON'T KNOW" // ถ้ามาถึงตรงนี้ คือไม่มีมากที่สุด (แสดงว่ามีมากที่สุดหลายอัน)



## 2 . Golden Rectangle by bT33

---

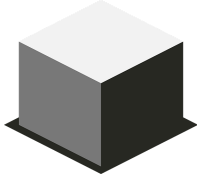
สำหรับโจทย์ข้อนี้ จะมีวิธีแก้ได้อยู่ด้วยกันหลายวิธี โดยในที่นี้จะขอแสดงวิธีเพียงวิธีเดียว

วิธีดังกล่าว คือ จะกำหนดให้สี่เหลี่ยมจัตุรัสรูปที่ 1 อยู่ที่ช่อง (0, 0) และในสี่เหลี่ยมจัตุรัสรูปถัด ๆ ไป จะคำนวณหาตำแหน่ง  $(r_i, c_i)$  ของช่องบนซ้ายและขนาดของรูปสี่เหลี่ยมจัตุรัสนั้น แล้วเลื่อนตำแหน่งของสี่เหลี่ยมจัตุรัสทุกรูปไป  $(-\min(r_i), -\min(c_i))$  ซึ่งจะทำให้ตำแหน่งซ้ายบนของสี่เหลี่ยมรูปใหญ่เป็น (0, 0) หลังจากนั้นให้นำข้อมูลตำแหน่งและขนาดเหล่านั้นมาสร้างรูปสี่เหลี่ยมตามต้องการในอาร์เรย์ 2 มิติ แล้วแสดงผลออกมาได้ตามที่ต้องการ

### เพิ่มเติม

จากการสังเกตเราจะพบว่า เมื่อให้  $S[n]$  = ความยาวด้านของสี่เหลี่ยมจัตุรัสรูปที่  $n$  เราจะพบว่า

$$S[n] = \begin{cases} 1 & \text{เมื่อ } n \leq 2 \\ S[n-1] + S[n-2] & \text{เมื่อ } n > 2 \end{cases}$$



### 3 . Alpha Number ! by JETHO

---

เราสามารถหาคำตอบได้โดยการสร้างเลขที่มีจำนวนหลักมากที่สุดซึ่งใช้เวลา  $O(V/\min(E_p))$

และใช้เวลาในการอัปเดตแต่ละหลักในเวลา  $O(10V/\min(E_p))$  โดยเราจะสร้างเลขหลักแรกจากเลขที่มากที่สุดที่ใช้พลังงานในการเขียนน้อยที่สุดโดยเลขนั้นจะต้องไม่เป็น 0 และสร้างเลขหลักอื่น ๆ จากเลขที่มากที่สุดที่ใช้พลังงานในการเขียนน้อยที่สุดที่รวมเลข 0 ด้วย

ด้วยวิธีดังกล่าวจะสามารถรับประกันได้ว่าจะได้เลขที่มีจำนวนหลักมากที่สุด จากนั้นเราจะอัปเดตตัวเลขแต่ละหลักให้เป็นตัวเลขที่มากที่สุดเท่าที่สามารถทำได้จากพลังงานที่เหลือ ประสิทธิภาพโดยรวมกรณีแย่ที่สุดคือ  $O(V)$  เมื่อ  $V$  คือ พลังงานเริ่มต้นที่คุณมี