

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 1. เกาะกลุ่ม (cluster)

เว็บวองไนพยายามแบ่งกลุ่มคนรีวิวร้านอาหารด้วยพื้นที่เพื่อหาว่าพื้นที่ไหนมีคนรีวิวเกาะกลุ่มกันเหนียวแน่นมากที่สุด เลยเก็บข้อมูลพิกัดคนรีวิวเป็นคู่ลำดับ  $(X,Y)$  โดย  $X$  แทนตำแหน่งแถว และ  $Y$  แทนตำแหน่งคอลัมน์ในพื้นที่สี่เหลี่ยมผืนผ้า หลังจากนั้นจึงทำการเกาะกลุ่มโดยวิธีง่ายๆคือ คนที่จะเกาะกลุ่มกันนั้นต้องมีคนรีวิวคนอื่นอยู่ติดกับตัวเองทิศใดทิศหนึ่งคือบน ล่าง ซ้าย ขวาอย่างน้อย 1 คน ยกตัวอย่างเช่น มีคนรีวิว 15 คน อยู่ที่ตำแหน่ง  $(0,0), (0,1), (0,3), (0,4), (0,5), (0,6), (1,2), (1,5), (2,0), (2,1), (2,4), (2,5), (3,0), (3,3), (3,5)$  หรือแสดงได้ดังแผนภาพต่อไปนี้

```
##.####
..#..#.
##.###
#..#.#
```

(ให้ # แทนตำแหน่งที่มีคนรีวิว และ . แทนตำแหน่งที่ไม่มีคนรีวิว ตำแหน่งเริ่มต้นคือ  $(0,0)$  ไข่มุมซ้ายบน)

ซึ่งจากพิกัดคนรีวิวนี้เราจะเห็นได้ว่าเราสามารถแบ่งกลุ่มคนรีวิวออกได้เป็น 5 กลุ่มดังรูป (ตัวเลขเดียวกันคือกลุ่มเดียวกัน)

```
11.2222
3..2.
44..22.
4..5.2.
```

ดังนั้นกลุ่มที่มีขนาดใหญ่ที่สุดคือ กลุ่มที่ 2 ซึ่งมีขนาดจำนวน 8 คน

ให้คุณเขียนโปรแกรมรับพิกัดของคนรีวิวแล้วหาคำตอบว่า กลุ่มคนรีวิวที่เกาะกลุ่มทางพื้นที่กันใหญ่ที่สุดมีจำนวนกี่คน

### ข้อมูลนำเข้า

บรรทัดแรก ค่า  $T$  ระบุจำนวน test case ที่จะทดสอบ

ในแต่ละชุดการทดสอบ

บรรทัดแรก จำนวนเต็มบวก  $M$   $N$  บอกจำนวนแถว และจำนวนคอลัมน์ของพื้นที่สำรวจ

บรรทัดที่สอง จำนวนเต็ม  $R$  แทนจำนวนคนรีวิวทั้งหมด

อีก  $R$  บรรทัดถัดมา แต่ละบรรทัด จำนวนเต็ม  $X_i$   $Y_i$  แทนพิกัดของคนรีวิวที่  $i$

### ข้อมูลส่งออก

บรรทัดเดียว ระบุขนาดของกลุ่มคนรีวิวที่มีขนาดใหญ่ที่สุด

### ขอบเขต

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

$$1 \leq T \leq 10$$

$$0 < M < 10$$

$$0 < N < 10$$

## ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
1 3 10 8 0 3 1 1 1 2 1 4 1 5 1 6 1 8 2 1	3

คำอธิบายตัวอย่าง

มี 1 เทสเคส

เทสเคสแรกและเทสเคสเดียว

ขนาด 3 แถว 10 คอลัมน์

มี 8 คนรีวิว

พิกัดคนจะได้ดังภาพ

...#.....

.##.###.#.

.#.....

จะได้กลุ่มทั้งหมด 4 กลุ่ม

...1.....

.22.333.4.

.2.....

กลุ่มที่ใหญ่สุดเท่ากัน 2 กลุ่ม คือ กลุ่มที่ 2 และ 3 โดยมีขนาดเป็น 3

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 2. หาคำหยาบแบบง่าย (Censor)

ในแต่ละวันมีผู้ใช้ Wongnai ที่เขียนรีวิวเข้ามาเป็นจำนวนมาก ทำให้เราต้องการระบบที่จะช่วยตรวจจับรีวิวที่มีการใช้คำหยาบอยู่ในเนื้อหา เพื่อที่จะ flag ไว้รอการตรวจสอบโดยแอดมินอีกครั้งหนึ่ง แต่ในระหว่างการรอตรวจสอบเราต้องการให้คำที่เป็นคำหยาบถูกเซ็นเซอร์ไปก่อน เพื่อไม่ให้ผู้ใช้คนอื่นเห็นคำหยาบเหล่านั้น

โจทย์ข้อนี้ เราต้องการให้เขียนโปรแกรมเพื่อทำการอ่านรีวิวของผู้ใช้ ค้นหาคำหยาบ (ที่ถูกกำหนดไว้แล้วในพจนานุกรมคำหยาบที่เตรียมไว้ให้) แล้วแทนที่ด้วยภาษาต่างดาว โดยภาษาต่างดาวคือ random string ที่ประกอบไปด้วยอักษร 8 ตัว ดังต่อไปนี้เท่านั้น

```
#]!(&@%?
```

ตัวอย่างภาษาต่างดาวที่มีความยาว 5 ตัวอักษร

```
#]!(&
```

ตัวอย่างภาษาต่างดาวที่มีความยาว 20 ตัวอักษร

```
#]!(&@%?#]!(&@%?#]!(
```

หมายเหตุ: ไม่ต้องสนใจเรื่องการตัดคำภาษาไทย เช่น “โหดเหี้ยม” ให้ถือว่าเป็นคำที่ต้องสงสัยว่าเป็นคำหยาบทันที เนื่องจากมีคำว่า “เหี้ย” อยู่ในคำ

### ข้อมูลนำเข้า (Input)

- ไฟล์พจนานุกรมคำหยาบ (rude-words.txt) ที่เป็น text file ที่มีคำหยาบ 1 คำต่อ 1 บรรทัด
- Text file ที่เนื้อหาเป็นรีวิวซึ่งอาจจะมีหรือไม่มีคำหยาบผสมอยู่ในนั้น
- ชื่อไฟล์ input จะเป็น 1.txt, 2.txt, 3.txt ไปจนถึง N.txt

### rude-words.txt (UTF-8 encoding)

```
เหี้ย  
หมาไม่แดก
```

### reviews.txt (UTF-8 encoding)

```
ร้านนี้อาหารเหี้ยมากๆ  
รสชาติหมาไม่แดกจริงๆ แดมเจ้าของกินนี้สยเหี้ยสุดๆ
```

# CODEHEW HACKATHON

---

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## ข้อมูลส่งออก (Output)

- Text file ที่มีชื่อไฟล์เป็น censored-N.txt เช่น censored-1.txt
- UTF-8 encoding
- ถ้า input file ไหนไม่มีคำหยาบ ไม่ต้องสร้างไฟล์ output ออกมา

## ตัวอย่างไฟล์ Output

```
ร้านนี้อาหาร#] ! (&มากๆ  
รสชาติ#] ! (&๕%?#จริงๆ แคมเจ้าของกินลี้#] ! (&สุดๆ
```

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 3. ห้องประชุม (Meeting room)

ปีนี้งานยังเต็มโตต่อเนื่องและได้ขยายทีมเพิ่มจำนวนมาก ใครชอบงานท้าทายสมัครงานกับเราได้ (ประกาศรับกันดื้อๆในใจทยอยเลยครับ :P) แต่ที่จะบอกก็คือ พอจำนวนคนเยอะขึ้นแล้วปัญหาสิ่งหนึ่งที่เกิดขึ้นคือเริ่มมีการประชุมของแต่ละทีมเพิ่มขึ้นตามไปด้วย โดยแต่ละทีมจะระบุเวลาที่จะเริ่มต้นใช้ห้องประชุม และเวลาเลิกใช้ห้องประชุมไว้ ซึ่งแต่เดิมนั้นแต่ละทีมก็เลือกใช้ห้องประชุมกันตามใจไม่มีระเบียบ ทำให้มีการใช้ห้องจำนวนมากเกินความจำเป็น บางห้องบางช่วงเวลาก็ไม่มีการใช้งาน ดังนั้นถ้าเราสามารถประหยัดจำนวนห้องประชุมได้ ก็จะสามารถเอาพื้นที่ห้องประชุมนั้นไปทำอย่างอื่น เช่น โต๊ะพูล หรือ โต๊ะปิงปอง เป็นต้น ดังนั้นจึงรบกวนคุณช่วยเขียนโปรแกรมหาหน่อยว่า จากตารางเวลาที่ใช้ห้องประชุมของแต่ละทีมนั้น จะใช้ห้องประชุมเป็นจำนวนน้อยที่สุดกี่ห้องที่ให้ทุกทีมประชุมในเวลาที่ต้องการได้

หมายเหตุ

1. ถ้าเวลาเลิกใช้ห้องประชุมของทีมหนึ่ง ตรงกับเวลาเริ่มประชุมของทีมหนึ่ง ไม่สามารถใช้ห้องประชุมเดียวกันได้
2. ทีมหนึ่งสามารถใช้เวลาเริ่มและเลิกเป็นเวลาเดียวกันได้

### ข้อมูลนำเข้า

บรรทัดแรก ค่า  $T$  ระบุจำนวน test case ที่จะทดสอบ

ในแต่ละชุดการทดสอบ

บรรทัดแรก จำนวนเต็มบวก  $N$  แทนจำนวนทีมที่จะใช้ห้องประชุม

อีก  $N$  บรรทัดถัดมา แต่ละบรรทัด จำนวนเต็ม  $B_i, E_i$  แทนเวลาเริ่มประชุม และเวลาสิ้นสุดของทีมที่  $i$

### ข้อมูลส่งออก

จำนวน  $T$  บรรทัด แต่ละบรรทัด ระบุจำนวนห้องที่น้อยที่สุดที่ต้องใช้ ของแต่ละชุดทดสอบ

### ขอบเขต

$$1 \leq T \leq 10$$

$$0 < N < 100$$

$$0 \leq B_i \leq E_i \leq 24$$

### ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
3	2
3	1
5 10	2
0 23	

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

15 20 4 1 2 3 4 5 6 7 8 2 18 18 18 19	
---	--

คำอธิบายตัวอย่าง

มี 3 เทสเคส

เทสเคส 1

มีจำนวน 3 ทีม

ทีม 1 เริ่มใช้ห้องเวลา 5 นาฬิกา เลิกใช้ 10 นาฬิกา

ทีม 2 เริ่มใช้ห้องเวลา 0 นาฬิกา เลิกใช้ 23 นาฬิกา

ทีม 3 เริ่มใช้ห้องเวลา 15 นาฬิกา เลิกใช้ 20 นาฬิกา

คำตอบคือ ใช้น้อยสุด 2 ห้อง ห้องแรกให้ทีมที่ 1 และทีมที่ 3 ส่วนห้องสองให้ทีมที่ 2

เทสเคส 2

มีจำนวน 4 ทีม

ทีม 1 เริ่มใช้ห้องเวลา 1 นาฬิกา เลิกใช้ 2 นาฬิกา

ทีม 2 เริ่มใช้ห้องเวลา 3 นาฬิกา เลิกใช้ 4 นาฬิกา

ทีม 3 เริ่มใช้ห้องเวลา 5 นาฬิกา เลิกใช้ 6 นาฬิกา

ทีม 3 เริ่มใช้ห้องเวลา 7 นาฬิกา เลิกใช้ 8 นาฬิกา

คำตอบคือ ใช้น้อยสุด 1 ห้อง เพราะไม่มีเวลาคาบเกี่ยวกันเลย

เทสเคส 3

มีจำนวน 2 ทีม

ทีม 1 เริ่มใช้ห้องเวลา 18 นาฬิกา เลิกใช้ 18 นาฬิกา

ทีม 2 เริ่มใช้ห้องเวลา 18 นาฬิกา เลิกใช้ 19 นาฬิกา

คำตอบคือ ใช้น้อยสุด 2 ห้อง ห้องแรกให้ทีมที่ 1 ส่วนห้องสองให้ทีมที่ 2

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 4. หารีวิวหน้าม้า (Bias reviews)

เว็บไซต์รีวิวที่เนื้อหาในเว็บไซต์ถูกเขียนขึ้นโดยผู้ใช้เป็นส่วนใหญ่ การควบคุมและตรวจจับรีวิวที่เป็นรีวิวหน้าม้า (ให้ 5 ดาว) หรือรีวิวกลับแง่ลบ (ให้ 1 ดาว) เป็นสิ่งที่สำคัญมาก แต่เนื่องจากจำนวนรีวิวในแต่ละวันมีจำนวนมาก ไม่สามารถตรวจสอบได้ด้วยแอดมินเพียงไม่กี่คน เราจึงต้องการคัดกรองรีวิวที่น่าสงสัยว่าจะเป็นรีวิวหน้าม้าหรือรีวิวกลับแง่ลบ แล้วจึงเริ่มตรวจด้วยทีมงานอีกครั้งหนึ่ง

ในโจทย์ข้อนี้ เราต้องการเครื่องคัดกรองเฉพาะกรณีที่เราสงสัยว่าจะเป็นรีวิวหน้าม้า โดยใช้เกณฑ์อย่างง่าย ๆ ดังต่อไปนี้

- เป็นการเขียนรีวิวครั้งแรกของผู้เขียนรีวิว (ผู้ใช้คนนี้อาจจะมีรีวิวทั้งหมดมากกว่า 1 รีวิว) และ
- เป็นรีวิวที่ให้ 5 ดาว และ
- ความยาวของรีวิวไม่เกิน 100 ตัวอักษร และ
- ผู้เขียนรีวิวมี Follower น้อยกว่า 100 คน

### ข้อมูลนำเข้า (Input)

ชุดของ Review และ User แยกเป็น 2 ไฟล์ (reviews.csv และ users.csv ตามลำดับ)

#### reviews.csv (UTF8 encoding)

column 1: Id ของรีวิว (โดยที่ค่าของ id ของรีวิวที่น้อยกว่า จะหมายถึงรีวิวที่เขียนก่อน)

column 2: Id ของ User

column 3: เรตติ้งที่เป็นตัวเลข 1-5 ดาว

column 4: เนื้อหาของรีวิว (เนื้อหาในรีวิวอาจมีอักขระขึ้นบรรทัดใหม่อยู่ด้วย)

ตัวอย่างไฟล์ reviews.csv

```
// review_id, reviewer_id, rating, review_text
2375, 55, 4, อาหารอร่อยที่สุดในสามโลก
9842, 8892, 5, ร้านนี้ให้ 5 ดาวเลย
723, 100, 5, ไม่มีร้านไหนแรงเท่าร้านนี้แล้ว
655, 35, 5, โคตรฟินครับร้านนี้
```

จากไฟล์ reviews.csv ข้างต้น จะหมายความว่า

รีวิว id = 2375 เขียนโดยผู้ใช้ที่มี id = 55 ให้ 4 ดาว และเนื้อหารีวิวคือ “อาหารอร่อยที่สุดในสามโลก”

รีวิว id = 9842 เขียนโดยผู้ใช้ที่มี id = 8892 ให้ 5 ดาว และเนื้อหารีวิวคือ “ร้านนี้ให้ 5 ดาวเลย”

รีวิว id = 723 เขียนโดยผู้ใช้ที่มี id = 100 ให้ 5 ดาว และเนื้อหารีวิวคือ “ไม่มีร้านไหนแรงเท่าร้านนี้แล้ว”

รีวิว id = 655 เขียนโดยผู้ใช้ที่มี id = 35 ให้ 5 ดาว และเนื้อหารีวิวคือ “โคตรฟินครับร้านนี้”

#### users.csv (UTF8 encoding)

column 1: Id ของ User

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

column 2: จำนวน Followers

ตัวอย่างไฟล์ users.csv

```
// user_id, number_of_followers
55,159
8892,65333
35,40
100,0
```

จากไฟล์ users.csv ข้างต้น จะหมายความว่า

User ที่มี id = 55 มี Followers จำนวน 159 คน

User ที่มี id = 8892 มี Followers จำนวน 65333 คน

User ที่มี id = 35 มี Followers จำนวน 40 คน

User ที่มี id = 100 มี Followers จำนวน 0 คน

## ข้อมูลส่งออก (Output)

- Textfile
- แต่ละบรรทัดเป็น id ของรีวิวที่สงสัยว่าจะเป็นรีวิวหน้าม้า (1 บรรทัดต่อ 1 id)
- เรียงลำดับข้อมูลในไฟล์ด้วยตัวเลขของ id

ตัวอย่างไฟล์ Output

```
655
723
```

หมายความว่า รีวิวที่มี id เท่ากับ 655, 723 เป็นรีวิวที่น่าสงสัยว่าจะเป็นรีวิวหน้าม้า



# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 5. จัดบู๊ท (Exhibition)

ในปี 2015 ที่ผ่านไป วงไนได้จัดงาน Bangkok Restaurant Week ซึ่งเป็นสุดยอดเทศกาลอาหาร และประสบความสำเร็จด้วยดีในปีนี้ก็เลยจะจัดอีกครั้ง เลยไปเช่าที่ Arena แห่งหนึ่งซึ่งได้พื้นที่ขนาดกว้าง  $M \times$  สูง  $N$  ตารางเมตรมา แล้วก็ไปสั่งจ้างบริษัทรับเหมาสร้างบู๊ทสำเร็จรูปให้ผู้ออกแสดงงานขนาด 4 ตารางเมตร โดยบอกไปว่าอยากได้เป็นรูปตัวไอ (I) คือ กว้าง 1 เมตร ยาว 4 เมตร เพื่อให้ใช้บล็อกหลังสุดเป็นที่เก็บของได้ แต่ว่าบริษัทผู้รับเหมาเข้าใจผิด ดันทำบล็อกมาเป็นรูปตัว L ด้านฐานตัว L กว้าง 2 เมตร ด้านสูงยาว 3 เมตร โดยฐานชี้ไปทางขวา เพราะนี่กว่าจะใช้ตรงปลายฐานตัว L เป็นที่เก็บของ

ฝ่ายจัดงานของวงไนปวดหัวมาก เพราะถ้าสั่งทำใหม่ก็ไม่ทันงานแล้ว เลยอยากรู้ว่าถ้าเป็นตัว L จะจัดได้รูปแบบไหนที่รูปแบบที่สามารถวางบู๊ทได้เต็มพื้นที่พอดี โดยให้ลำดับการวางบู๊ทที่แตกต่างกันถือเป็นรูปแบบที่แตกต่างกันแม้ว่าผลลัพธ์สุดท้ายจะหน้าตาเหมือนกัน

### ข้อมูลนำเข้า

บรรทัดแรก ค่า  $T$  ระบุจำนวน test case ที่จะทดสอบ

ในแต่ละชุดการทดสอบ

บรรทัดเดียวระบุ จำนวนเต็มบวกสองจำนวน  $M$   $N$  แทนขนาดของพื้นที่

### ข้อมูลส่งออก

จำนวน  $T$  บรรทัด แต่ละบรรทัด ระบุจำนวนรูปแบบที่เป็นไปได้ ของแต่ละชุดทดสอบ

### ขอบเขต

$$1 \leq T \leq 10$$

$$0 < M, N < 7$$

### ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
2	2
2 4	0
2 3	

# CODEHEW HACKATHON

---

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## คำอธิบายตัวอย่าง

มี 2 เทสเคส

เทสเคสแรก พื้นที่กว้าง 2 สูง 4 สามารถวางได้ 2 รูปแบบคือ (ให้ตัวเลขแทนลำดับชั้นที่วาง)

11

21

21

22

และ

22

12

12

11

นั่นคือถึงแม้ทั้ง 2 แบบนี้จะได้ผลลัพธ์การวางออกมาเหมือนกัน แต่นับว่าวางได้ 2 รูปแบบ

เทสเคสสอง พื้นที่กว้าง 2 สูง 3

ไม่สามารถวางให้เต็มพื้นที่ได้

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 6. ขอบเขตแม่/ลูก (Intersect)

เนื่องจากใน Wongnai มีข้อมูลขอบเขตของเขตและจังหวัดอยู่จำนวนมาก โดยแต่ละเขตก็就会被เก็บให้มีความสัมพันธ์กับจังหวัดที่เป็น parent

โดยขอบเขตของพื้นที่เหล่านี้ถูกวาดขึ้นโดยแอดมิน แอดมินจะค่อยๆ วาด polygon บนแผนที่เพื่อสร้างขอบเขตของจังหวัดเป็น polygon ก่อน แล้วจึงวาด polygon อื่นๆ เพื่อสร้างขอบเขตของเขตต่างๆ ในจังหวัดนั้น

ปัญหาที่แอดมินเจอคือ ในขณะที่วาด polygon ของเขตอยู่ มันจะมีโอกาสที่จะวาดพื้นที่ของเขตเกินออกนอกกรอบของจังหวัดที่เขตนั่นอยู่

ทำให้ทีมงานจำเป็นต้องสร้างเครื่องมือที่ไว้ช่วยแก้ไข polygon ที่วาดเกินพื้นที่ของ parent ให้ถูกต้อง โดยจะแก้ไข polygon ของเขต ให้ snap ไปกับ polygon ของจังหวัดนั้นให้อัตโนมัติ

โจทย์ข้อนี้ต้องการให้ช่วยสร้างโปรแกรมที่รับ polygon ของเขตและจังหวัดไป แล้วคำนวณหา polygon ของเขตแต่ละเขตที่ถูกตัด ไม่เกินออกจากพื้นที่ที่เป็น parent ของมัน

### ข้อมูลนำเข้า (Input)

จุดเริ่มต้นของจุดแรกจะเป็นจุดที่ใกล้จุด (0, 0) มากที่สุด และจุดลำดับต่อไป จะวนตามเข็มนาฬิกา

บรรทัดที่ 1: เซ็ตของจุดของ polygon ที่เป็นจังหวัด

บรรทัดที่ 2: เซ็ตของจุดของ polygon ที่เป็นเขต

### รูปแบบ Input

x11, y11 x12, y12 x13, y13 x14, y14

x21, y21 x22, y22 x23, y23 x24, y24

โดยที่ค่า x,y จะเป็น double หรือ int ก็ได้

### ตัวอย่างไฟล์ Input

```
0,0 100,0 100,100 0,100
50,50 125,50 125,125 50,125
```

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย



รูปแสดงตัวอย่างไฟล์ input และ output

- พื้นที่สีแดง เป็นพื้นที่ของจังหวัด
- พื้นที่สีเขียวอ่อน เป็นพื้นที่ของเขตที่ถูกวาด
- พื้นที่สีเขียวเข้ม เป็นพื้นที่ของเขตที่ควรจะเป็น (หลังจากถูก Snap)

## ข้อมูลส่งออก (Output)

- Textfile
- บรรทัดที่ 1: จำนวนจุดของ polygon ของพื้นที่เขตว่ามีกี่จุด
- บรรทัดที่ 2: เชื่อมจุดของ polygon ของเขตที่ snap เข้ากับ พื้นที่ของ parent แล้ว (แต่ละจุดให้เว้นวรรค 1 วรรค)
- จุดเริ่มต้นของจุดแรกต้องเป็นจุดที่ใกล้จุด (0, 0) มากที่สุด และจุดลำดับต่อไป ให้วนตามเข็มนาฬิกา
- ความถูกต้องระดับทศนิยม 7 ตำแหน่ง (ปัดทศนิยมจากตำแหน่งที่ 8,  $\geq 5$  ปัดขึ้น)
- เมื่อ snap เข้ากับพื้นที่ของ parent แล้ว จำนวนจุดจะต้องน้อยที่สุด
- หากพื้นที่ของเขต ไม่ทับกับพื้นที่ของจังหวัดเลย ให้ใส่แค่จำนวนจุดเป็น 0

รูปแบบของ Output

```
numberOfPoints  
x1,y1 x2,y2 x3,y3 x4,y4 x5,y5
```

ตัวอย่างไฟล์ Output

```
4  
50,50 100,50 100,100 50,100
```

จากไฟล์ Output จะได้เชื่อมของจุด polygon ของเขตหลังทำการ snap กับ parent เป็น 4 จุด ดังนี้ (50, 50) , (100, 50) , (100, 100) , (50, 100)

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 7. ลูกเทพ (Thep child)

คุณเทพ ประกอบธุรกิจร้านอาหารมาแล้วหลายร้าน แต่ปรากฏว่าล้มเหลวเกือบทุกร้าน มีวิธีไว้ในเว็บวงในก็ดีตลอดเลยไม่ทราบสาเหตุว่าเพราะอะไรถึงเจ๊งกันแน่ จึงได้ปรึกษาคุณลูก (ลูกเทพ) ที่กำลังศึกษาวิชาไซสศาสตรอยู่ ลูกเทพจึงมาช่วยคุณพ่อ

ลูกเทพนั่งดูปฏิทินการเปิดปิดร้านอาหารแต่ละร้านของคุณพ่อแล้วรู้สึกอะไรบางอย่างมีความคล้ายคลึงกันนั่นคือ ถ้าเป็นชาวบ้านปกติ จะปิดร้านทุกวันอาทิตย์ แต่คุณพ่อเทพ กลับเลือกที่จะปิดร้านสลับสับเปลี่ยน เช่น ปิดวันที่ 20, 31, 1, 2, 3, 4, 7, 12, 15, 21, 3, 4, 7, 12, 15, 21, 3, 4, 7, 12, 15, 21, 3, 4,...

แต่ลูกเทพก็สังเกตเห็นรูปแบบอะไรบางอย่างนั่นคือ ลำดับวันที่ปิดร้านนั้นเกิดลูปวนซ้ำขึ้น นั่นคือ 3, 4, 7, 12, 15, 21 เกิดซ้ำไปเรื่อยๆ และทุกๆร้านที่เจ๊งก็มีลูปวนซ้ำในลักษณะเดียวกัน ลูกเทพจึงฟันธงว่าร้านจะเจ๊ง

ถ้ามีลำดับวันที่การปิดร้านเกิดลูปวนซ้ำขึ้นนั่นเอง

คุณพ่อเทพเห็นดังนั้น

จึงนำวันปิดร้านย้อนหลังของร้านใหม่ที่กำลังดำเนินการอยู่มาให้ลูกเทพช่วยดูให้ว่าร้านจะเจ๊งเพราะมีลำดับวันที่การปิดร้านเป็นลูปวนซ้ำหรือไม่

หมายเหตุ ในทุกชุดการทดสอบ ถ้าเป็นชุดที่มีลูปจะการันตีว่าจะมีลูปอย่างน้อย 2 รอบ เมื่อเกิดลูปแล้วจะเกิดลูปไปเรื่อยๆ และตัวเลขในลูปจะไม่ซ้ำกัน แต่อาจจะซ้ำกับตัวเลขนอกลูปหรือไม่ก็ได้

### ข้อมูลนำเข้า

บรรทัดแรก ค่า T ระบุจำนวน test case ที่จะทดสอบ

ในแต่ละชุดการทดสอบ

บรรทัดแรก ระบุจำนวนเต็มบวก N แทนจำนวนวันที่มีการปิดร้าน

บรรทัดที่สอง ระบุจำนวนเต็มบวก N จำนวน แต่ละจำนวนคือเลขวันที่ที่มีการปิดร้าน

### ข้อมูลส่งออก

จำนวน T บรรทัด แต่ละบรรทัด ตอบ yes ถ้าร้านจะเจ๊ง หรือ ตอบ no ถ้าร้านไม่เจ๊ง

### ขอบเขต

$1 \leq T \leq 10$

$0 < N < 1000$

$1 \leq \text{เลขวันที่ปิดร้าน} \leq 31$  (สมมติว่าให้คิดว่าทุกเดือนมี 31 วันเพื่อความสะดวก)

### ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
2	no
3	yes
1 3 5	

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

12	
1 3 5 2 3 1 2 3 1 2 3 1	

## 8. หารูปซ้ำ (Duplicate Photos)

ในแต่ละวัน มีผู้ใช้ Wongnai โปสรูปเข้ามาในระบบจำนวนหลายพันรูปทั้งจาก Desktop และจากมือถือ ซึ่งโดยปกติแล้วคนที่โปสรูปก็จะเป็นเจ้าของรูปนั่นเอง แต่ก็เสี่ยงไม่ได้ถ้าจะมีรูปบางรูปที่ถูกคัดลอกแล้วนำมาโพสต์ใหม่โดย User อีกคน เพื่อให้รีวิวของตนเองมีรูปภาพประกอบ และทำให้ได้รีวิวนั้นตรงตามเกณฑ์การได้เป็นรีวิวแบบคุณภาพ (Quality Review)

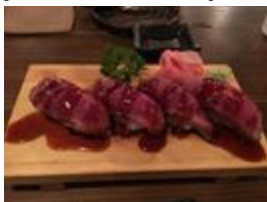
เราให้ความสำคัญกับปัญหาการละเมิดลิขสิทธิ์รูปภาพ จึงต้องมีโปรแกรมไว้ตรวจสอบการซ้ำกันของรูปภาพในระบบ เพื่อทีมงานจะได้นำไปตรวจสอบอย่างละเอียดต่อไปว่าเป็นการคัดลอกมาจริงหรือไม่

โจทย์ข้อนี้จะให้เขียนโปรแกรมที่ไว้ตรวจสอบรูปภาพจากชุดของรูปภาพจำนวน 1,000 รูป ที่กระจายอยู่ในหลายๆ folder

โดยกำหนดให้กรณีของภาพที่ซ้ำกันมีดังต่อไปนี้



- รูปเดียวกันที่ทุก Pixel เหมือนกันหมด ไม่มีจุดที่แตกต่าง
- รูปเดียวกันแต่ขนาดรูปไม่เท่ากัน (ความกว้าง หรือ ความยาว ไม่เท่ากัน --- รูปต้นฉบับถูก resize รูป)



# CODEHEW HACKATHON

---

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

- รูปเดียวกันที่ถูกหมุนรูป Rotate clockwise/counter-clockwise



- รูปเดียวกันที่ถูก flip แนวนั่ง หรือ แนวนอน



# CODEHEW HACKATHON

---

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

- รูปเดียวกันแต่ถูกนำมาปรับค่า Contrast หรือค่า Brightness



- รูปเดียวกันแต่ปรับให้สีเป็น Grayscale



- รูปเดียวกันแต่มีการใส่ Watermark เข้าไปเพิ่มเติม เช่น กรณีคนเซฟรูปของคนอื่นที่ไม่ได้ใส่ watermark ไว้ เอาไปใส่ watermark ของตัวเองลงไป แล้วมาโพสต์ใหม่





# CODEHEW HACKATHON

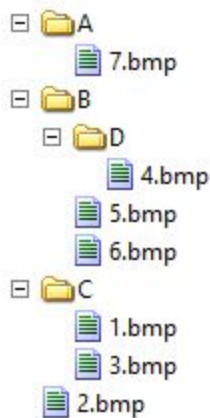
โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

- รูปเดียวกันแต่โดน crop รูป เพื่อตัดข้อมูลบางส่วนของรูปออกไป เช่น การนำภาพที่มี Watermark มาโพสใหม่ แต่ทำการ crop ส่วนที่มี Watermark ออกไป)



## ข้อมูลนำเข้า (Input)

- ไฟล์รูปภาพ Bitmap (.bmp) ที่อยู่ใน folder ต่างๆ จำนวน N ไฟล์ (โปรแกรมจะต้องทำการ traverse ไปตาม sub-folder เพื่อหาไฟล์ .bmp เอาเอง) โดยชื่อไฟล์ที่อยู่ตาม folder ต่างๆ จะไม่มีไฟล์ที่ชื่อซ้ำกัน



หมายเหตุ: ในชุดของรูปภาพที่กรรมการจะนำมาทดสอบ

- รูปที่ซ้ำกัน จะซ้ำกันเป็นคู่เท่านั้น นั่นคือ ถ้ารูป 1.bmp ซ้ำกับ 9.bmp แล้ว จะไม่มีรูปอื่นที่ซ้ำกับ 1.bmp และ 9.bmp อีก
- การซ้ำกันของรูป จะไม่ได้มาจากการใช้ 2 เทคนิคพร้อมกัน

## ข้อมูลส่งออก (Output)

- ผลลัพธ์เป็น Text file
- แต่ละบรรทัดเป็นรายชื่อไฟล์ของรูปที่ซ้ำกัน คั่นด้วย comma (,) และตัดนามสกุลของไฟล์ออก
- ในแต่ละบรรทัด ชื่อไฟล์รูปที่ซ้ำ จะต้องเรียงลำดับตัวเลขน้อยไปหามาก

# CODEHEW HACKATHON

---

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

- เรียงลำดับบรรทัดในไฟล์ ด้วยการใช้ชื่อไฟล์ของไฟล์แรกในแต่ละบรรทัด เป็นตัวจัดเรียงจากน้อยไปหามาก

ตัวอย่างไฟล์ Output

2, 9 17, 675
-----------------

จากไฟล์ผลลัพธ์ข้างบนนี้ จะหมายความว่า

รูป 2.bmp, 9.bmp ซ้ำกัน

รูป 17.bmp, 675.bmp ซ้ำกัน

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 9. กี่ร้าน? (How many?)

ในระบบแผนที่แบบ grid มีการเก็บว่าที่ตำแหน่งพิกัด (x,y) ใหนมีร้านอยู่บ้าง แต่เนื่องจากแผนที่ไม่ได้ละเอียดมากจึงอาจมีหลายร้านอยู่ที่พิกัดเดียวกันก็ได้ นักวิเคราะห์ข้อมูลของวงในมักจะอยากรู้ข้อมูลว่า ในอาณาบริเวณสี่เหลี่ยมที่ต้องการมีจำนวนร้านอยู่ทั้งหมดกี่ร้าน โดยจะระบุจุดมุมซ้ายบน (x,y) ความกว้าง w และ ความสูง h ของสี่เหลี่ยม จึงต้องการโปรแกรมที่ช่วยหาคิวรีจำนวนร้านให้ได้อย่างรวดเร็ว

ใน 1 เทสเคสนั้นมีได้หลายคิวรี ให้ตอบเป็นผลรวมของทุกคิวรีรวมกัน

หมายเหตุ ให้จุด (0,0) คือจุดซ้ายล่าง

### ข้อมูลนำเข้า

บรรทัดแรก จำนวนเต็มบวก T แทนจำนวนเทสเคส

แต่ละเทสเคสประกอบด้วย

บรรทัดแรก จำนวนเต็มบวก 3 จำนวน N W H

โดย N แทนจำนวนร้านทั้งหมดที่มีในแผนที่

W แทนความกว้างของแผนที่

H แทนความสูงของแผนที่

อีก N บรรทัดถัดมา แต่ละบรรทัดระบุจำนวนเต็ม 2 จำนวน x y คือพิกัดของร้านแต่ละร้าน

บรรทัดถัดไป จำนวนเต็มบวก Q ระบุจำนวนคิวรีที่นักวิเคราะห์ข้อมูลจะถาม

อีก Q บรรทัดถัดมา แต่ละบรรทัดระบุจำนวนเต็ม 4 จำนวน x y w h

โดย x และ y แทนพิกัดซ้ายบนของพื้นที่สี่เหลี่ยมที่ต้องการหาคิวรี

w และ h แทนความกว้าง และ ความสูงของสี่เหลี่ยม

### ข้อมูลส่งออก

จำนวน T บรรทัด แต่ละบรรทัดระบุจำนวนผลรวมของทุกคิวรีของเทสเคสนั้นรวมกัน

### ขอบเขต

$1 \leq T \leq 10$

$1 \leq W, H \leq 1,000$

$1 \leq N \leq 1000$

$1 \leq Q \leq 1,000,000$

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
1 10 5 6 1 1 4 3 2 3 3 3 4 3 1 1 2 3 4 0 0 3 4 5 4 0 2 5 3 2 3 2 2 3 3 4 4 1 5 4 5	18

## คำอธิบาย

มี 1 test case

พื้นที่ขนาดกว้าง 5 สูง 6

มีจำนวนร้านอาหาร 10 ร้านในพื้นที่ดังนี้

```
0 0 0 0 1
0 0 0 0 0
1 0 2 1 2
0 0 0 0 0
0 2 0 0 0
0 0 0 0 1
```

คิวรีแรก จุดซ้ายบน (0,2) กว้าง 5 สูง 3 มีร้านทั้งหมด 3 ร้านดังนี้

```
0 0 0 0 1
0 0 0 0 0
1 0 2 1 2
0 0 0 0 0
0 2 0 0 0
0 0 0 0 1
```

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

คิวรีสอง จุดขายบน (2,3) กว้าง 2 สูง 2 มีร้านทั้งหมด 3 ร้านดังนี้

0	0	0	0	1
0	0	0	0	0
1	0	2	1	2
0	0	0	0	0
0	2	0	0	0
0	0	0	0	1

คิวรีสาม จุดขายบน (3,3) กว้าง 4 สูง 4 มีร้านทั้งหมด 4 ร้านดังนี้

0	0	0	0	1
0	0	0	0	0
1	0	2	1	2
0	0	0	0	0
0	2	0	0	0
0	0	0	0	1

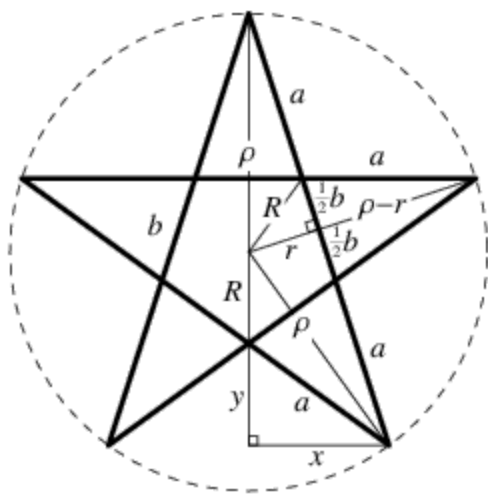
คิวรีสี่ จุดขายบน (1,5) กว้าง 4 สูง 5 มีร้านทั้งหมด 8 ร้านดังนี้

0	0	0	0	1
0	0	0	0	0
1	0	2	1	2
0	0	0	0	0
0	2	0	0	0
0	0	0	0	1

ดังนั้นคำตอบคือ  $3 + 3 + 4 + 8 = 18$

## 10. ร้านไหนอยู่ในรูปดาว (Star boundary)

มีนักคณิตศาสตร์เทพคนหนึ่งเจอวิธีค้นหาสถานที่จากตำแหน่งปัจจุบันที่จะทำให้ได้ร้านที่น่าพึงพอใจมากที่สุดแบบใหม่ โดยเค้าพบว่าการค้นหาร้านแบบเดิมที่กวาดรัศมีออกไปเป็นวงกลมจากตำแหน่งปัจจุบัน ได้ผลไม่ดีเท่ากับการวาดรูป star polygon (ดาว 5 แฉก <http://mathworld.wolfram.com/Pentagram.html>) โดยให้จุดศูนย์กลางของดาว ตรงกับตำแหน่งปัจจุบัน โดยนักคณิตศาสตร์คนนี้ได้เค้าอ้างว่าร้านไหนก็ตามที่อยู่ในพื้นที่รูปดาวนี้ จะมีโอกาสสูงมากที่จะเป็นร้านที่ผู้ใช้พึงพอใจจนอยากเดินทางไปลองชิม



กลุ่มโปรแกรมเมอร์สายแดงที่รักการกินเป็นชีวิตจิตใจกลุ่มหนึ่งต้องการพิสูจน์วิธีการค้นหาแบบใหม่นี้ แล้วก็ตกลงว่าจะลองเปลี่ยนไปใช้วิธีใหม่ดู โดยจะค้นหาร้านในพื้นที่รูปดาว 5 แฉกแทน ทำให้ต้องเขียนโปรแกรมเพื่อตรวจสอบว่าร้านที่เลือกขึ้นมาั้น อยู่ในบริเวณรูปดาว 5 แฉกหรือไม่

เราคือโปรแกรมเมอร์กลุ่มนั้น ฮ่าๆ

### ข้อมูลนำเข้า (Input)

Text file ที่มีเนื้อหาในไฟล์ตามนี้

- บรรทัดที่ 1
  - ตำแหน่ง X,Y
- บรรทัดที่ 2
  - ระยะจากจุดศูนย์กลางถึงจุดที่ไกลที่สุดของดาว (เพื่อกำหนดขนาดของดาว)
- บรรทัดที่ 3 เป็นต้นไปจนจบไฟล์ จะเป็นพิกัด X, Y ที่ต้องการให้ตรวจสอบ

ตัวอย่างไฟล์ Input

# CODEHEW HACKATHON

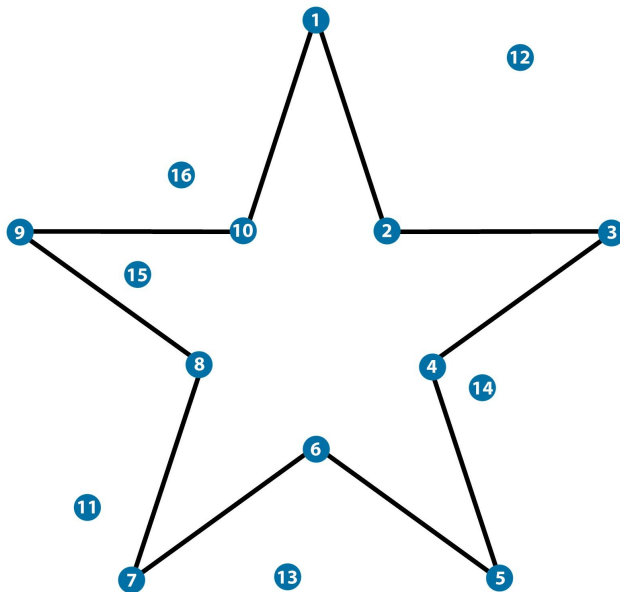
โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

```
13.7173454,100.5790912
0.5
13.3590796,99.2759945
13.9949196,101.1756706
13.9949196,101.1756706
13.995444,100.443889
13.438658,100.6515435
13.4387555,100.818313
```

จากไฟล์ input หมายความว่า ให้ตรวจสอบว่าตำแหน่ง 6 ตำแหน่งข้างล่างนี้ อยู่ในดาว 5 แฉกที่มีจุดศูนย์กลางอยู่ที่ 13.7173454,100.5790912 หรือไม่

## ข้อมูลส่งออก (Output)

- Text file
- บรรทัดที่ 1 ถึง 10 คือ จุดของ Polygon รูปดาว 5 แฉก (ทศนิยมต้องถูกต้อง 7 ตำแหน่ง) โดยเรียงลำดับจุดตามรูปดาวข้างล่างนี้ (เฉพาะจุดที่ 1 ถึงจุดที่ 10)



- บรรทัดที่ 11 เป็นต้นไป จะเป็นค่า 0 หรือ 1 เพื่อบอกว่าพิกัดที่ให้ตรวจสอบ อยู่ภายในดาว 5 แฉกหรือไม่ โดยที่
  - ค่า 0 หมายถึง พิกัดนั้นไม่อยู่ในดาว

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

- ค่า 1 หมายถึง พิกัดนั้นอยู่ในดาว

ตัวอย่างไฟล์ Output

```
13.7173454,101.0790912
13.8296022,100.7335999
14.1928738,100.7335999
13.8989807,100.5200742
14.0112381,100.174583
13.7173454,100.3881085
13.4234527,100.174583
13.5357101,100.5200742
13.241817,100.7335999
13.6050886,100.7335999
0
0
0
0
1
0
```

จากไฟล์ Output จะหมายความว่า

มีพิกัดที่ 5 (13.438658, 100.6515435) เพียงพิกัดเดียวที่อยู่ในดาว 5 แฉกที่มีจุดศูนย์กลางอยู่ที่ 13.7173454, 100.5790912 และมีความยาวจากจุดศูนย์กลางถึงจุดที่ไกลสุดของดาวเท่ากับ 0.5



# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 11. จอมเวทย์ (The great magician)

จอมเวทย์เวนต์หลงทางไปพบหอสมุดโบราณที่มีตำราเวทย์เก็บไว้เป็นจำนวนมาก  
ว่ากันว่าตำราเวทย์ใดมีข้อความคาถาอาคมซ้ำกันยิ่งยาว ก็ยิ่งมีพลังมาก เวนต์ไม่สามารถศึกษาตำรามหาศาลทั้งหมดได้  
จึงต้องการเลือกเฉพาะเล่มที่ให้พลังแก่ร่างเท่านั้น

โปรแกรมเมอร์เอ๋ย จะช่วยเวนต์เลือกตำราทรงพลังออกมาจากหอสมุดได้หรือไม่

### ข้อมูลนำเข้า

บรรทัดแรก จำนวนเต็ม  $N$  แทนจำนวนตำราเวทย์

บรรทัดต่อไป เป็นข้อความในตำราเวทย์แต่ละเล่ม (ASCII) ข้อความในตำราเวทย์แต่ละเล่มจะสิ้นสุดด้วยข้อความว่า `finem libri.` โดยไม่จำเป็นต้องขึ้นบรรทัดใหม่หรือเว้นวรรค และจะไม่มีข้อความว่า `finem libri.`

อยู่ในเนื้อหาของตำราแต่ละเล่ม จะปรากฏเป็นคำปิดเท่านั้น

ทั้งนี้ข้อความในตำราเวทย์แต่ละเล่มอาจประกอบด้วยหลายบรรทัด

การซ้ำกันของข้อความในตำราเวทย์ที่ให้พลังแก่ร่างได้นั้นจะต้องซ้ำไปจนถึง white space ในข้อความด้วย เช่นถ้า

ข้อความ

america กับ

a

m

e

r

i

c

a

ไม่ถือว่าซ้ำกันเนื่องจากข้อความหลังประกอบด้วย new line character คั่นระหว่างตัวอักษรต่างๆ

### ข้อมูลส่งออก

$N$  บรรทัด บรรทัดที่  $i$  บอกความยาวของข้อความที่ยาวที่สุดที่มีซ้ำกันภายในตำราเล่มที่  $i$

### ขอบเขต

ข้อความในหนังสือแต่ละเล่มมีความยาวสูงสุด 12,000 ตัวอักษร

ข้อความสามารถสะกดด้วยตัวอักษรเล็กหรือใหญ่หรือสลับกันก็ได้ ถือว่าเป็นคำเดียวกัน (Case insensitive)

### ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
--------------	--------------

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

4	16
Spirit Magic or Celestial Spirit Magic can summon celestial spirits by opening their gates.Finem libri.	19
A type of Magic which grants the user various characteristics belonging to a Sky Dragon, allowing them to incorporate the element of air into their body. However, unlike other types of Dragon Slayer Magic, Sky Dragon Slayer Magic is also geared more towards healing and support rather than just offense. The healing power granted by it is extremely high, having been described as capable of healing wounds inflicted by a real Dragon. Finem libri.	11
Dragon Force greatly increases the damage done by standard Dragon Slayer spells, and grants the user access to more advanced, powerful attacks. It also drastically enhances the user's physical prowess, making them far stronger, more resistant to damage, and faster, and considerably boosts their Magic Power, which becomes visible, taking the appearance of a Dragon. When exuded at its fullest, said Magic Power takes on a violent corporeality, likely as a result of the immense amount of Magic Power employed. When Natsu Dragneel entered this state through the use of Jellal Fernandes' Flame of Rebuke, his flames gained the ability to burn off offending Magic, even that of Zero's ultimate attack Genesis Zero. Finem libri. An extremely powerful ability that allows two Mages to unite two kinds of Magic similar in nature together, thereby creating an even stronger attack. According to Jellal Fernandes, many priests have wasted their	12

# CODEHEW HACKATHON

---

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

entire lives trying to achieve Unison Raid without success, thus implying that it is an extremely difficult ability to master, or let alone obtain. However, Lucy Heartfilia and Juvia Lockser succeeded on their first attempt, as did Wendy Marvell. Finem libri.	
---	--

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 12. หาผู้ใช้ที่ให้การให้เรตติ้ง มีค่า S.D. มากที่สุด (S.D.)

เนื่องจากผู้ใช้แต่ละคนใน Wongnai มีเกณฑ์การให้คะแนนร้านที่ต่างกัน รวมถึงความยากง่ายของการให้คะแนนของแต่ละคนก็แตกต่างกัน ผู้ใช้บางคนให้คะแนนกระจายมาก มีทั้งให้ 1 ดาว 3 ดาว หรือ 5 ดาว หรือบางคนก็ให้คะแนนอยู่กลางๆ สะส่วนใหญ่

เราต้องการวิเคราะห์พฤติกรรมการให้คะแนนของผู้ใช้แต่ละคน ทำให้เราอยากเห็นค่าเบี่ยงเบนมาตรฐาน S.D. (Standard deviation) ของการให้ดาวของผู้ใช้แต่ละคน เพื่อมาวิเคราะห์เบื้องต้น

แต่สิ่งที่เกิดอยากรู้ขึ้นมาก็คือ ผู้ใช้ท่านไหนกันนะที่ให้ดาวร้านต่างๆ ได้แปรปรวนมากที่สุด โดยที่เราจะดูเฉพาะผู้ใช้ที่มีรีวิวมากกว่า 10 รีวิวเท่านั้น

สิ่งที่ต้องการคือ id ของผู้ใช้ที่มีลักษณะการให้คะแนนแปรปรวนมากที่สุด และต้องการเห็นค่า S.D. การให้คะแนนของผู้ใช้คนนี้

### ข้อมูลนำเข้า (Input)

#### reviews.csv

ไฟล์ที่มี reviews ทั้งหมดในระบบ (UTF-8 encoding)

column 1: Id ของรีวิว (โดยที่ค่าของ id ของรีวิวที่น้อยกว่า จะหมายถึงรีวิวที่เขียนก่อน)

column 2: Id ของ User

column 3: เรตติ้งที่เป็นตัวเลข 1-5 ดาว

ตัวอย่างไฟล์ reviews.csv

```
// review_id, reviewer_id, rating
182,1116,4
188,1117,2
```

โดยข้อมูลรีวิวนี้นจะมีทั้งข้อมูลของคนที่มีร่วมน้อยกว่า 10 รีวิว และมากกว่าเท่ากับ 10 รีวิวรวมอยู่ด้วย

### ข้อมูลส่งออก (Output)

บรรทัดที่ 1: Id ของ user ที่ให้ดาวได้แปรปรวนมากที่สุด

บรรทัดที่ 2: ค่า S.D. ของการให้ดาวของ User ในบรรทัดที่ 1

ความถูกต้องของค่า S.D. จะต้องถูกต้องระดับทศนิยม 4 ตำแหน่ง (ปัดทศนิยมจากตำแหน่งที่ 5)

# CODEHEW HACKATHON

---

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

ตัวอย่างไฟล์ Output

55 1.7051
--------------

จากไฟล์ Output หมายความว่า User ที่มี Id = 55 มีค่า S.D. เป็น 1.7051

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 14. หาค่า MD5 hash ของไฟล์ขนาดใหญ่ (MD5)

ต้องการจะหา MD5 hashing ของไฟล์ๆ หนึ่ง ซึ่งมีลักษณะของ content ดังนี้

ไฟล์เป็น text file (utf8 encoding) ที่ประกอบไปด้วยตัวอักษร 1-0a-zA-Zก-ฮ (หรือ 1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZกขคดขงจฉชฌญฎฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮ) วนต่อกันไปเรื่อยๆ ให้มีขนาดยาวที่สุดแต่ไม่เกิน 2GB (2\*1024\*1024 bytes)

### Note

- ถ้าหากตัวอักษรตัวสุดท้ายทำให้จำนวน bytes เกินขนาดที่กำหนด ทุก bytes ของตัวอักษรนั้นจะถูกตัดทิ้งไป
- สามารถดูการเรียงพยัญชนะภาษาไทยได้ที่ <https://th.wikipedia.org/wiki/อักษรไทย>

### ข้อมูลส่งออก (Output)

ค่า MD5 hash ของไฟล์ข้างต้น (โดยค่า A-F ให้ใช้ตัวพิมพ์เล็กทั้งหมด)

### ตัวอย่างไฟล์ Output

```
b41f787a17edf62c82f60a66463ff5b2
```

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## 15. ไร้ปัญญาประดิษฐ์ (A.I. - Artificial Ignorance)

เมื่อวันที่ 27 มกราคม 2559 ที่ผ่านมา Google ได้ประกาศว่าปัญญาประดิษฐ์ (A.I. - Artificial Intelligence) สำหรับเล่นหมากล้อม หรือ โกะ (Go) ของตัวเองที่ชื่อว่า AlphaGo นั้น สามารถชนะแชมป์หมากล้อมยุโรป ได้ 5 กระดานติดต่อกัน ซึ่งนับว่าเป็น breakthrough สำหรับงานด้าน A.I. เป็นอย่างมาก

ทางทีมงานอ่านข่าวเลยเกิดแรงบันดาลใจสร้างขึ้นมาบ้าง แต่เนื่องจากว่าเกมหมากล้อมยากไป ทีมงานทำไมเป็น เลยได้สร้าง A.I. สำหรับเล่น OX (โอเอ็กซ์) ขึ้นมา โดยจะให้ผู้เล่นแข่งสร้าง A.I. ขึ้นมาแข่งกับ A.I. ของทีมงาน (สำหรับคนที่ไม่รู้จัก OX อ่านวิธีเล่น OX ในหมายเหตุท้ายข้อนะครับ)

แต่ว่า A.I. ของทีมงานนั้นยังง่อยอยู่ โดยมีอัลกอริทึมในการเล่นดังนี้

1. ถ้าพบว่าตำแหน่งไหนว่างหมากลึงไปแล้วชนะทันที จะลงที่ตำแหน่งนั้น
2. ถ้าไม่มีตำแหน่งไหนที่จะชนะ แต่พบว่ามีการแข่งจะชนะได้ทันที จะวางหมากลึงไปป้องกันที่ตำแหน่งนั้น
3. กรณีอื่นๆ คิดไม่ออกกว่ายังไงดี เลยจะสุ่มตำแหน่งว่างๆ ตำแหน่งหนึ่งว่างหมากลึงไป

จะเห็นว่า A.I. ของเรานั้นไม่ค่อยฉลาดเท่าไร จะประดิษฐ์ให้เก่งก็ไร้ปัญญา (เลยตั้งชื่อว่า “ไร้ปัญญา(จะ)ประดิษฐ์” นั่นเอง) คุณเห็นแล้วสงสัยเลยจะต่อให้โดยให้ทุกเกม A.I. ของทีมงานจะเริ่มเดินก่อนเสมอ โดยทีมงานเลือกหมาก O เสมอ ส่วนคุณใช้หมาก X

เราจะแข่งกัน 5 กระดาน แต่ละกระดานถ้าคุณชนะจะได้ 2 คะแนน เสมอได้ 1 คะแนน และแพ้ได้ 0 คะแนน

ส่วนวิธีเขียน A.I. มาแข่งกับเราคือ จะต้องมีการเขียนโปรแกรมเพื่อยิง POST request ไปยัง A.I. server ของเรา โดยมีขั้นตอนดังนี้

1. เริ่มเล่นเกม ให้โปรแกรมของคุณยิง POST request ไปที่ `http://128.199.205.4:3000/ai/play` โดยให้ใส่ header เป็น “Content-type: application/json” และ data เป็น json ในรูปแบบ

```
{
  "team": "<team_name>"
}
```

โดย <team\_name> ให้ระบุชื่อทีมของผู้แข่งขัน  
แล้วรอรับ response กลับมา

2. A.I. Server จะส่ง response มาในรูปแบบ json เช่นกัน คือ

```
{
  "status": <status_code>,
  "table": "<current_table>"
}
```

# CODEHEW HACKATHON

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

โดย <status\_code> เป็น 0 หมายถึงเกมยังดำเนินอยู่

เป็น 1 หมายถึง จบเกม ฝ่าย O ชนะ

เป็น 2 หมายถึง จบเกม ฝ่าย X ชนะ

เป็น 3 หมายถึง จบเกม เสมอ

เป็น -1 หมายถึง error ซึ่งอาจเกิดจากไม่ได้ส่งชื่อทีมไป

<current\_table> เป็น string ความยาว 9 ตัวอักษร ระบุสถานะตารางปัจจุบันหลังจาก A.I.

ได้ทำการวางหมากแล้ว โดย 3 ตัวอักษรแรกคือแถวบนสุดของตาราง 3 ตัวอักษรถัดมาคือแถวกลาง

และ 3 ตัวอักษรสุดท้ายคือแถวล่างสุด โดยให้ space (ช่องว่าง) หมายถึงตำแหน่งที่ยังไม่มีหมาก

เช่น "XX O XO XO" คือตาราง

X	X	
O		X
O	X	O

ซึ่งแน่นอนว่าถ้าเป็นตาเดินแรก ใน string 9 ตัวอักษรจะมี O อยู่ตำแหน่งเดียว ที่เหลือเป็น space

### 3. โปรแกรมคุณเมื่อได้รับ response แล้ว ให้ทำงานดังนี้

ถ้า status\_code เป็น -1, 1, 2, 3 ให้จบการทำงานของโปรแกรมทันที

ถ้า status\_code เป็น 0 ให้คำนวณช่องที่จะวางหมากตำแหน่งถัดไป แล้วยิง POST request ไปที่ <http://128.199.205.4:3000/ai/move> โดยให้ใส่ header เป็น "Content-type: application/json" และ data เป็น json คือ

```
{
  "team": "<team_name>",
  "table": "<table_after_move>"
}
```

โดย <team\_name> คือ ชื่อทีม

<table\_after\_move> คือ string ความยาว 9 ตัวอักษร

แทนสถานะของตารางใหม่หลังจากได้วางหมากไปแล้ว

และรอรับ response โดยการทำงานจะวน ข้อ 2 และ 3 ไปจนกว่าจะจบกระดาน

### วิธีการตรวจ

ข้อนี้เวลาตรวจ ให้เดินมาที่โต๊ะกรรมการเพื่อแจ้งว่าทำการตรวจ และจะเริ่มการแข่งขัน 5 เกม

แต่ละเกม จะให้ผู้แข่งขันส่งทำงานโปรแกรม โดยเริ่มที่ขั้นตอนที่ 1 และทำงานส่วนที่เหลือโดยอัตโนมัติ

และรอผลสรุปแพ้ชนะจาก A.I. Server (ถ้าเกิด error เนื่องจากมีการส่งข้อมูลไม่ตรงตามรูปแบบ จะถือว่าแพ้ทันที)

หลังจากนั้นกรรมการจะตรวจสอบขั้นตอนการเดินว่าไม่มีอะไรผิดปกติ พอจะขึ้นเกมใหม่ ก็จะให้รันโปรแกรมใหม่อีกครั้ง

จนครบ 5 เกม



# CODEHEW HACKATHON

---

โดย Wongnai & สมาคมโปรแกรมเมอร์ไทย

## Test และ Judge Server

ในการทดสอบระบบให้ใช้ URL เป็น 128.199.205.4:8080 สามารถทดสอบกี่ครั้งก็ได้

ในการตรวจคะแนนให้ใช้ URL เป็น 128.199.205.4 (port 80)

โดยห้ามยิงมาที่เครื่องตรวจคะแนนจนกว่าจะมาตรวจที่โต๊ะกรรมการ มิฉะนั้นจะถือว่าได้ 0 คะแนน

## หมายเหตุ

วิธีเล่น OX (ใครเล่นเป็นแล้วไม่จำเป็นต้องอ่าน)

มีตารางขนาด 3x3 จะผลัดกันลงหมาก ฝ่ายหนึ่งเป็น O ฝ่ายหนึ่งเป็น X ลงไปที่ช่อง ถ้าฝ่ายไหนสามารถให้เรียง 3 หมากของตัวเองติดกันได้ ไม่ว่าจะเป็นแนวนอน แนวตั้ง หรือ แนวทแยง จะเป็นผู้ชนะเกมไป แต่ถ้าลงจนครบทุกช่องแล้วไม่มีฝ่ายไหนเรียงติดกัน 3 หมากได้ จะถือว่าเสมอ

เช่น

O		X
	O	X
		O

แบบนี้ถือว่า O ชนะ เพราะเรียงต่อกันในแนวทแยง