[1] Sorting Acronyms

เขียนโปรแกรมเรียงชื่อ ตาม “ชื่อแบบอักษรย่อ” ของชื่อเต็ม ที่รับเข้ามาจาก Standard Input โดยมีหลักการย่อคำ ดังนี้

1. ชื่อเต็ม จะประกอบด้วยคำภาษาอังกฤษหลายคำ
2. การสร้างชื่อย่อ จะใช้ตัวอักษรตัวแรกของแต่ละคำในชื่อเต็ม ในกรณีที่ตัวอักษรแรกนั้นๆ เป็นตัวอักษร พิมพ์ใหญ่
3. กรณีที่ตัวอักษรตัวแรกเป็นตัวพิมพ์เล็ก จะไม่นำมาประกอบในชื่อย่อ

ตัวอย่างเช่น United States of America ย่อว่า USA เช่นเดียวกับ the United States of America แต่หากเป็น The United States Of America จะย่อว่า TUSOA และ University of Tsukuba ย่อว่า UT แต่หากเป็น University Of Tsukuba จะย่อว่า UOT เป็นต้น

Input (อ่านจาก Standard Input)

บรรทัดแรกของ Input จะเป็นตัวเลขจำนวนเต็ม N ที่แสดงถึงจำนวนชื่อที่มีใน Input ทั้งหมด และจากนั้น N บรรทัด จะเป็นชื่อแต่ละชื่อที่จะต้องทำการย่อและนำไปเรียง

Output (แสดงผลใน Standard Output)

ให้แสดงผลชื่อย่อของชื่อเต็มที่รับเข้ามา บรรทัดละ 1 ชื่อ โดยเรียงลำดับจากชื่อที่ยาวที่สุด ไปยังชื่อที่สั้นที่สุด โดยหาก มีความยาวเท่ากัน ให้เรียงตามลำดับอักษร (Alphabetical Order)

ตัวอย่าง Input & Output

Input	**Output**
5 the United States of America The United States of America Carnegie Mellon University The United States Of America the united states	TUSOA TUSA CMU USA A

[2] Floating Prime

นิยาม Floating Prime (แบบช้าๆ): สำหรับตัวเลขทศนิยมที่มีค่าระหว่าง 1.0 - 10.0 ใดๆ จะเป็น Floating Prime เมื่อแปลงตัวเลขทีละชุดจากตัวเลขหน้าจุดทศนิยม ไปจนถึงทศนิยมตำแหน่งที่ 1, 2, และ 3 ให้เป็นจำนวนเต็ม หากตัว ที่แปลงแล้วตัวใดตัวหนึ่งเป็นจำนวนเฉพาะ ให้ถือว่าตัวเลขทศนิยมนั้นๆ เป็น Floating Prime ..... งงล่ะสิ :P งั้นดู
ตัวอย่าง

1.43318374 **เป็น** Floating Prime เพราะ
พิจารณาถึงทศนิยมตำแหน่งที่ 1 ได้ 14 ไม่ใช่จำนวนเฉพาะ
พิจารณาถึงทศนิยมตำแหน่งที่ 2 ได้ 143 ไม่ใช่จำนวนเฉพาะ
พิจารณาถึงทศนิยมตำแหน่งที่ 3 ได้ 1433 เป็นจำนวนเฉพาะ

1.6172746483 **ไม่เป็น** Floating Prime เพราะ
พิจารณาถึงทศนิยมตำแหน่งที่ 1 ได้ 16 ไม่ใช่จำนวนเฉพาะ
พิจารณาถึงทศนิยมตำแหน่งที่ 2 ได้ 161 ไม่ใช่จำนวนเฉพาะ
พิจารณาถึงทศนิยมตำแหน่งที่ 3 ได้ 1617 ไม่ใช่จำนวนเฉพาะ

1.31234567 **เป็น** Floating Prime เพราะ
พิจารณาถึงทศนิยมตำแหน่งที่ 1 ได้ 13 เป็นจำนวนเฉพาะ

ให้เขียนโปรแกรมรับตัวเลขทศนิยม แล้วบอกว่าเป็น Floating Prime หรือไม่

Input (อ่านจาก Standard Input)
รับตัวเลขทศนิยม (รวมตัวเลขหน้าจุดแล้วยาวไม่เกิน 12 ตำแหน่ง) ทีละหนึ่งตัว และนำไปพิจารณาว่าเป็น Floating Prime หรือไม่ โปรแกรมจะต้องทำงานไปจนกระทั่งรับตัวเลข 0.0 ถึงจะหยุดทำงาน

Output (แสดงผลใน Standard Output)
แสดงข้อความ TRUE ในกรณีที่เป็น Floating Prime และ FALSE ในกรณีที่ไม่เป็น

ตัวอย่าง Input & Output

Input	Output
1.43318374 1.31234567 1.6172746483 0.0	TRUE TRUE FALSE

[3] Digit Hangman

เกม Digit Hangman เป็นเกมเลียนแบบเกม Hangman แต่มีกติกาที่เปลี่ยนไปเล็กน้อย คือ ใช้ตัวเลข 12 หลัก (กำหนดตายตัว) แทนการใช้คำศัพท์และการเดาจะเดาเป็นตัวเลข 0-9 และจะต้องเดาทั้งหมด 5 ครั้ง (ตายตัวเช่นเดียวกัน) และมีวิธีการนับคะแนนคือ จำนวนหลักของโจทย์ที่สามารถเปิดออกมาได้ (ดังนั้นถึงจะเดาถูกเหมือนกันก็อาจได้คะแนนไม่เท่ากัน เช่น ตัวเลขในโจทย์มี 3 อยู่ 5 ตัว และ 1 อยู่ 2 ตัว ถ้าเดา 3 ถูกก็จะได้ 5 คะแนน ในขณะที่ 1 จะได้ 2 คะแนน)

ทั้งนี้ถึงจะสามารถเดาถูกหมดภายในการเล่นไม่ถึง 5 ครั้ง (ซึ่งเป็นไปได้) ก็จะต้องเล่นให้ครบ 5 ครั้ง ซึ่งจะไม่มีผลใดๆ ทั้งสิ้นกับคะแนน เพราะคะแนนจะนับจากตัวเลขที่เปิดได้เท่านั้น แต่ก็ยังถือว่าเป็นการเดาที่ผิด และต้องแสดงตัวเลขที่เดาผิดอยู่ (ดูตัวอย่างสุดท้ายในตัวอย่างข้อมูลนำเข้า/ผลลัพธ์ด้านล่าง)

ให้เขียนโปรแกรมเพื่อจำลองการเล่น Digit Hangman

Input (อ่านจาก Standard Input)

บรรทัดแรก คือโจทย์ของเกม เป็นตัวเลขจำนวนเต็ม 12 หลัก แต่ละหลักคั่นด้วยช่องว่าง และ 5 บรรทัดต่อมา คือการ เดาแต่ละครั้ง เป็นตัวเลขจำนวนเต็มระหว่าง 0-9 บรรทัดละ 1 ตัว

Output (แสดงผลใน Standard Output)

มี 7 บรรทัด โดยบรรทัดแรกเป็นโจทย์ตอนเริ่มเล่น จากนั้นแต่ละบรรทัดจะเป็นผลการเล่นจากการเดาแต่ละครั้ง โดยแต่ละหลักคั่นด้วยช่องว่าง และตัวเลขที่เคยเดาผิดไปแล้ว ตามลำดับการเดา แต่ละตัวคั่นด้วยช่องว่าง บรรทัดสุดท้ายจะเป็นคะแนนรวมทั้งหมดที่ผู้เล่นทำได้นับจากจำนวนหลักทั้งหมดที่เปิดมาได้ (เต็ม 12 คะแนน)

ตัวอย่าง Input & Output

ตัวอย่างข้อมูลนำเข้า	**ผลลัพธ์**
4 4 3 5 6 0 6 6 0 0 4 6 3 2 1 6 0	_ _ _ _ _ _ _ _ _ _ _ _ _ _ 3 _ _ _ _ _ _ _ _ _ _ _ 3 _ _ _ _ _ _ _ _ _ 2 _ _ 3 _ _ _ _ _ _ _ _ _ 2 1 _ _ 3 _ 6 _ 6 6 _ _ _ 6 2 1 _ _ 3 _ 6 0 6 6 0 0 _ 6 2 1 8
0 6 6 3 8 4 3 2 9 6 7 8 9 0 5 6 8	_ 9 _ _ _ 0 _ _ _ _ _ _ _ 9 _ _ _ 0 _ _ _ _ _ _ _ 9 _ _ _ 5 0 6 6 _ _ _ _ _ 9 6 _ _ 5 0 6 6 _ 8 _ _ _ 9 6 _ 8 5 7

ตัวอย่างข้อมูลนำเข้า	ผลลัพธ์
9 9 4 2 2 4 7 7 9 6 6 4 1 5 8 3 0	_ 1 _ _ _ _ _ _ _ _ _ _ _ _ 1 5 _ _ _ _ _ _ _ _ _ _ _ _ 1 5 8 _ _ _ _ _ _ _ _ _ _ _ _ 1 5 8 3 _ _ _ _ _ _ _ _ _ _ _ _ 1 5 8 3 0 0
9 9 4 2 2 4 7 7 9 6 6 4 9 6 7 2 4	_ _ _ _ _ _ _ _ _ _ _ _ 9 9 _ _ _ _ _ _ 9 _ _ _ 9 9 _ _ _ _ _ _ 9 6 6 _ 9 9 _ _ _ _ 7 7 9 6 6 _ 9 9 _ 2 2 _ 7 7 9 6 6 _ 9 9 4 2 2 4 7 7 9 6 6 4 12
2 2 2 2 2 3 3 3 3 3 3 2 3 4 5 6	_ _ _ _ _ _ _ _ _ _ _ _ 2 2 2 2 2 _ _ _ _ _ _ _ 2 2 2 2 2 3 3 3 3 3 3 2 2 2 2 2 3 3 3 3 3 4 2 2 2 2 2 3 3 3 3 3 4 5 2 2 2 2 2 3 3 3 3 3 4 5 6 12

หมายเหตุ: น้องๆ อาจจะคิดว่ากฎและข้อจำกัดของเกมนี้ไม่สมเหตุผลเอาเสียเลย เขียน Hangman ธรรมดาดีกว่า ขอให้ตระหนักว่า “โจทย์ในชีวิตจริงของการเขียนโปรแกรม มันไม่ได้สมเหตุผลทุกเรื่องหรอก มีกรณีมากมายที่เราจะเจอ Requirement อะไรแปลกๆ แบบนี้”

[4] Basic Web Crawler

ทีม Marketing โครงการ Internship ต้องการทำ API สำหรับ Logo บริษัทที่เข้ามาร่วมในโครงการ แต่เนื่องจากไม่อยากรบกวนทีม Developer ของโครงการ (ที่แต่ละคนเป็นอาสาสมัครจากบริษัทที่เข้าร่วมโครงการ ที่งานล้นมืออยู่แล้ว) จึงคิดวิธีการง่ายๆ ที่จะ “ดึงเฉพาะ Logo URL มาจากหน้าเว็บของโครงการ The Internship” และขอให้น้องๆ ช่วยเขียนโปรแกรมนี้

โจทย์ข้อนี้มี 2 ส่วน โดยที่น้องๆ สามารถทำเฉพาะส่วนแรก (4.1) ก็ได้แต่ถ้าอยากได้คะแนนเพิ่ม ก็ทำส่วนที่สอง (4.2) ด้วยจะเยี่ยมมากเลย ..... แต่ส่วนที่สองจะทำได้ต้องทำส่วนแรกได้ก่อนนะ :-)

[4.1] Extract Data from Source HTML

1. เขียนโปรแกรมเพื่อดึงข้อมูล URL ของ Logo ทุกบริษัท ที่อยู่ในส่วน “ใครมาบ้าง / PARTICIPATING STARTUPS” ในเว็บไซต์ The Internship (https://theinternship.io/) โดยข้อมูลต้นทางจะเป็น HTML และมีข้อมูลที่เป็น URL ของ Logo แต่ละบริษัทอยู่

2. ให้ Output ข้อมูลโดยเรียงลำดับตามความยาวของ “คำอธิบายบริษัท” ซึ่งอยู่ใต้รูป โดยเรียงจากน้อยไปมาก

3. ถ้ามีการเปลี่ยนแปลงจำนวนของบริษัทบนหน้าเว็บ (มีบริษัทเข้ามาใหม่ หรือ ออกจากโครงการ) เมื่อรันโปรแกรมใหม่ผลลัพธ์จะต้องเปลี่ยนไปตามข้อมูลบนหน้าเว็บจริง

ตัวอย่าง Output

company/wisible_logo.png
company/codeapp_logo.png
company/horganice_logo.png
…

[4.2 - Optional] Extract Data from Source HTML

ให้น้องๆ เอาข้อมูลที่ได้มาจากข้อ 4.1 มาเปิดเป็น JSON API สำหรับเข้าถึงข้อมูล โดยมี Spec ดังนี้

- มี Route คือ /companies และรองรับเฉพาะ GET Request
- เมื่อเรียก API แล้วจะได้ JSON ที่มีโครงสร้างดังตัวอย่าง

```
{ "companies" : [
        { "logo": "https://theinternship.io/company/wisible_logo.png"},
        { "logo": "https://theinternship.io/company/codeapp_logo.png"}
] }
```

น้องๆ สามารถใช้ภาษาโปรแกรมไหนก็ได้, Framework ไหนก็ได้ที่ถนัด หรืออยากใช้ :-)