Programming for NLP
PA 2: Hangman

คำชี้แจง

Programming Assignment นี้มีสองส่วน ส่วนแรกจะเป็นโจทย์สั้น ๆ 2 โจทย์เพื่อให้ได้ฝึกการใช้ String และ Dictionary เพิ่มเติม ส่วนที่สองจะให้เขียนเกมทายคำ (เกมส์ Hangman)

เมื่อเสร็จแล้วให้ส่ง **word_guess.py** บน mcv โดย**ห้ามแก้ไขชื่อไฟล์โดยเด็ดขาด**

การตรวจจะตรวจโดยอัตโนมัติด้วย autograder เพื่อทดสอบว่ามันรันได้ ไม่ติด syntax error และผ่าน test case แบบง่าย ๆ แต่ว่าจะมี test case อื่นอีกที่ไม่ได้เปิดให้เห็น นอกจากนั้น autograder จะตรวจความคล้ายคลึงของงานกับเพื่อนคนอื่นๆ โดยอัตโนมัติและบอกทันทีว่างานเราคล้ายกับใครบ้าง บรรทัดไหน นับเป็นกี่ percent ถ้าเครื่องตรวจเจอจะถูกปรับเป็น 0 สำหรับ PA นี้ เพราะฉะนั้นคุยกับเพื่อนได้ แต่ต้องเขียน solution ของเราเอง และแต่ละคนมักจะไม่เหมือนกันเลย

- ชิ้นงานนี้มีการเก็บคะแนน 10 คะแนน
- ระยะเวลาการทำ 14 วัน (5 กันยายน - 20 กันยายน 23:59 น.) กรณีส่งสาย ชิ้นงานจะถูกตัดคะแนน 20% ทันที
- หากมีข้อสงสัยหรือปัญหาระหว่างการทำ assignment สามารถสอบถามทาง Discord ได้

เกมทายคำ

เราจะเขียนโปรแกรมทายคำ (เกม Hangman) เป็นเกมง่ายๆ ตรงไปตรงมาที่คนสมัยพี่ๆ ลุงๆ เล่นกันบนกระดาษ ก่อนที่ทุกคนจะเริ่มเล่นเกมแบบสวยๆ ฟริ้งๆ ผ่านมือถือ กติกาคือผู้เล่นจะแค่รู้ว่าคำศัพท์นั้นมีกี่ตัวอักษร แต่ต้องเดาทีละตัว แล้วเกมจะบอกว่าตัวอักษรนั้นมีหรือไม่มี ถ้ามีมีตรงไหนในคำ ผู้เล่นจะทายไปเรื่อยๆ จนกว่าจะทายถูกครบทั้งคำก็จะชนะ แต่ถ้าทายผิดเกิน 8 ครั้งก็จะแพ้

ตัวอย่างการรันโปรแกรม

```
The word now looks like this: -----

You have 8 guesses left.

Type a single letter here, then press enter: a

That guess is correct.


The word now looks like this: -A---

You have 8 guesses left.
```

Your past guesses are A.

Type a single letter here, then press enter: q

There are no Q's in the word.

The word now looks like this: -A---

You have 7 guesses left.

Your past guesses are AQ.

Type a single letter here, then press enter: pp

A guess should be a single character.

The word now looks like this: -A---

You have 7 guesses left.

Your past guesses are AQ.

Type a single letter here, then press enter: p

That guess is correct.

The word now looks like this: -APP-

You have 7 guesses left.

Your past guesses are AQP.

Type a single letter here, then press enter: C

There are no C's in the word.

The word now looks like this: -APP-

You have 6 guesses left

Your past guesses are AQPC.

Type a single letter here, then press enter: H

That guess is correct.

```
The word now looks like this: HAPP-
You have 6 guesses left.
Your past guesses are AQPCH.
Type a single letter here, then press enter: y
That guess is correct.
Congratulations, the word is: HAPPY
```

ถ้าเกิดผู้เล่นเล่นจนแพ้

```
...
The word now looks like this: HAPP-
You have 1 guess left.
Your past guesses are AQPCHZMNTJ.
Type a single letter here, then press enter: L
There are no L's in the word.
Sorry, you lost. The secret word was: HAPPY
```

โปรแกรมนี้เราจะ decompose ออกมาให้ทำเป็นสองส่วน คือส่วนของ gameplay และส่วนของการดึงคำมาจากไฟล์คำศัพท์

PART 1. ส่วนหลักของเกมและกฎกติกาของเกม (function playgame)

ในส่วนนี้ให้เรา assume ไปก่อนว่า get_word() จะ return คำปริศนาที่ใช้สำหรับทาย สิ่งที่เราต้องทำคือ

- print บอกผู้เล่นว่ารูปร่างของคำเป็นยังไงบ้าง ดังนี้
  The word now looks like this: -APP-
- print บอกผู้เล่นว่าเหลือโอกาสทายผิดอยู่เท่าไร ดังนี้
  You have 6 guesses left. หรือ You have 1 guess left.
- print บอกผู้เล่นว่าเดาอะไรไปแล้วบ้าง (ทั้งผิดและถูกตามลำดับ) ดังนี้
  Your past guesses are AQPC.(ไม่ต้องทำรูปเอกพจน์)
- ขอ input จากผู้เล่นว่าอยากจะเดาตัวอักษรอะไร

    - ถ้าผู้เล่นเดามามากกว่าหนึ่งตัวอักษร ให้เตือนบอกว่าต้องใส่มาแค่ตัวอักษรตัวเดียว แต่ไม่ถือว่าเป็นการตอบผิดในตานั้น ดังนี้
      A guess should be a single character.
    - เช็คว่าเดามาถูกหรือไม่ และอัพเดทสถานะให้เหมาะสม
- ถ้าผู้เล่นทายถูกทั้งคำแล้วให้บอกว่าชนะแล้ว ดังนี้

  Congratulations, the word is: HAPPY
- ถ้าผู้เล่นทายผิดครบจำนวนครั้งที่กำหนดให้บอกว่าแพ้ และเฉลยคำที่ถูกต้อง
  Sorry, you lost. The secret word was: HAPPY

<u>รายละเอียดสำคัญเมื่อส่งงาน</u>

- การตรวจความถูกต้องของโปรแกรมดูจากสิ่งที่โปรแกรม print ออกมาใส่ console เพราะฉะนั้น program ที่ส่งควรจะ print ออกมาให้เหมือนตัวอย่างที่สุดเท่าที่จะทำได้
  (เช่น upper/lower case)
- ต้องใช้ค่า INITIAL_GUESSES ในการกำหนดจำนวนตาเล่นตอนเริ่มเกมส์ใหม่ ไม่เช่นนั้น autograder จะตรวจไม่ถูก เพราะ autograder จะเปลี่ยนจำนวน INITIAL_GUESSES เพื่อทดสอบว่าโปรแกรมที่ส่งมาใช้ค่านี้ในการกำหนดจำนวนตาเล่นหรือไม่

PART 2. ดึงเอาคำปริศนามาจากไฟล์

ตอนนี้ get_word ดึงคำปริศนามาจาก list สั้นๆ ซึ่งทำให้เกมไม่สนุกน่าเบื่อ ในส่วนนี้เราจะเขียน get_word ใหม่โดยให้ดึงคำปริศนามาจากไฟล์ที่เก็บไว้ในตัวแปร LEXICON_FILE และสุ่มคำออกมาหนึ่งคำเป็นคำปริศนา

ใน starter code มี test lexicon ให้ด้วยให้ใช้สำหรับการทดสอบ **แล้วก็อย่าลืม .strip() ก่อนนำคำมาใช้ ไม่งั้น \n จะมาโผล่ในคำซึ่งทำให้ทายยังไงก็ไม่ถูก**

1. Using `.readlines()`

This reads **all lines at once** into a list.

python Copy code

```python
# Open the file
with open("example.txt", "r") as f:
    lines = f.readlines()

print(lines)  # Each line is an item in the list
```

Example output if the file contains:

nginx Copy code

```
Harry
Ron
Hermione
```

Result:

python Copy code

```python
['Harry\n', 'Ron\n', 'Hermione\n']
```

คำถามที่พบบ่อย (FAQ)

- ไม่ต้องคิดกรณีที่ทายตัวอักษรซ้ำหลายครั้ง => ปล่อยผ่าน
- หากใส่จุด เว้นว่าง ตัวเลข ภาษาอื่น => ปล่อยผ่าน
- การแสดงผลต้องออกมาเป็นประโยคเหมือนตัวอย่างเป๊ะๆ เช่น เหลือหลาย guess ก็ใช้ guesses เหลือ 1 guess ก็ใช้ guess เฉยๆ ใส่จุด เว้นวรรคให้เหมือนตัวอย่าง เพราะใช้ autograder ตรวจ