

stanCode

標準程式教育機構

Assignment 3

This assignment is based on the Assignment 2 of CS106AP and the Assignment 3 of CS106A at Stanford University



作業檔案下載

這份作業將帶領同學熟悉 CS 領域中最重要的基本觀念：**string** 並運用 **string manipulation** 的技巧來解決生物科技的問題、密碼學問題、與大人小孩都喜愛的「吊死鬼」單字遊戲。請注意：這次作業每一題請自己建造 **function(s)** 來達到 **decomposition**。如果有做 **extension**，請繳交兩份檔案：一份是沒有 **extension** 但達到所有作業的要求的，另一份是加入 **extension** 並請更改檔名為“原本的檔名_ext.py”

估計需要時間為 15 小時

如果作業卡關 歡迎與助教討論，也非常鼓勵同學們互相討論作業之 概念，但請勿把 **code** 給任何人看（也請不要把程式碼貼在社團裡），分享您的 **code** 會剝奪其他學生獨立思考的機會，也會讓其他學生的程式碼與你的極度相似，使防抄襲軟體認定有抄襲嫌疑

Problem 1 - rocket.py

這一題需要各位同學幫忙，使用 **print** & **double for loop** 打造台灣新型火箭！
（如下圖所示）



做出一架火箭需要專業分工，可以拆成六個部分（如下圖所示）：

- 火箭頭 (head)
- 聯結 (belt)
- 火箭上半身 (upper)
- 火箭下半身 (lower)
- 聯結 (belt)
- 火箭頭 (head)

每一個部分都是一個 **function**，且每個 **function** 裡 至少 需要一個 **for loop(s)**

更具體地說，**def main()** 裡面不應該出現任何的 **print**，而是許多的 **functions** 例如 **head()**, **belt()**, **upper()**, **lower()**, **belt()**, **head()**，就好像是 Karel 作業時大家

請勿使用上課未討論過的 **Python** 指令（例如：文字的乘法）



Problem 2 - complement.py

Problem 2 和 3 將帶領同學了解 `coding` 在生物科技上的應用 - DNA 序列工程

DNA 是由四種含氮鹼基 A (Adenine), T (Thymine), C (Cytosine), G (Guanine) 所構成。A 與 T 為互補鹼基序列、C 與 G 為互補鹼基序列（互補鹼基序列：能量最低、最穩定的鹼基對結構）。如下圖所示：



請編輯 `complement.py` 讓使用者可以輸入一個 DNA 片段並得到該片段之最低能量互補片段。您完成的程式應該要能完美重現下圖中的所有文字

```
Please give me a DNA strand and I'll find the complement: ATGCAAG
The complement of ATGCAAG is TACGTTC
```

學會 `function`, `parameters` 與 `return value` 的您應該要能使用 **decomposition** 的概念讓程式碼架構更清晰。請在 `complement.py` 裡建造一個名為 `build_complement()` 的 `function` 可以 `return` 出輸入其中的互補序列。舉例來說，`build_complement('ATC')` 應該要 `return` 出 `'TAG'`

請注意，`build_complement` 應該是 **case-insensitive**，也就是 `build_complement('atC')` 與 `build_complement('Atc')` 都會得到相同的結果 `'TAG'`

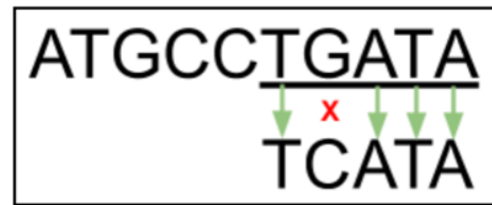
（您可以假設使用者輸入的 DNA 片段裡只會包含 `'a', 'A', 't', 'T', 'c', 'C', 'g', 'G'`）

Problem 3 - similarity.py

coding 在生物產業界很常見的應用就是找出某較短 DNA 片段在較長的 DNA 片段中配對率最高的區間，我們稱之 **homology**



相似率：40%



相似率：80%

假設使用者輸入的 DNA 序列為 "ATGCCTGATA"(以下稱之為 `long_sequence`)、待配對的 DNA 片段為 "TCATA"(以下稱之為 `short_sequence`)，我們可以從 `long_sequence` 裡取一個跟 `short_sequence` 一樣長的連續片段比對，並得到在 `long_sequence` 裡與 `short_sequence` 相似度最高區間(如上圖所示，配對率最高的片段為 "TGATA")

您的程式完成時將可以完美複製下圖中的所有文字：

```
Please give me a DNA sequence to search: ACTGACATTG
What DNA sequence would you like to match? TGCCA
The best match is TGACA
```

注意：您的程式應該是 **case-insensitive**。也就是使用者輸入的文字大小寫不會影響最後得到的答案(如下圖所示)

```
Please give me a DNA sequence to search: ATcgAtCGatCgC
What DNA sequence would you like to match? tCgC
The best match is TCGC
```

Problem 4 - caesar.py

加密 (Cipher) 是非常普遍的保密手段，甚至有許多數學家或是電腦科學家潛心研究這門學問 — 密碼學 (Cryptography)。電影「模仿遊戲」(The Imitation Game) 故事劇情主軸就是天才數學家 兼 密碼學家 兼 電腦科學家 的圖靈 (Turing) 試圖解密 - decipher - 由英國軍隊所攔截的德國海軍情報

這題將帶領同學了解簡易的密碼學，並讓同學們熟悉 string manipulation

Cryptography 這門學問非常有趣，最早的源頭竟然要追溯到西元前 70 年！羅馬共和國獨裁者 — 凱薩 發明了最早的加密系統 “Caesar Cipher” 去保護他的機密不讓第三方知道

大家熟知的英文字母排列順序如下圖所示 (未平移 的英文字母串我們稱之為 **ALPHABET**)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

如果我們把 **ALPHABET** 所有英文字母向右平移 4 格，超過最後一個字母的，把它放回第一個 (wrap around)，如此一來我們就可以得到一個全新的英文字母串 **new_alphabet**：

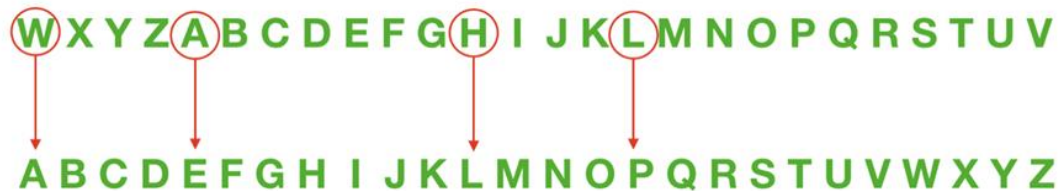
W X Y Z A B C D E F G H I J K L M N O P Q R S T U V

到底要怎麼使用 **new_alphabet**？舉個例子，今天 Caesar 想要傳一個機密訊息 “APPLE” 給 Jerry，Caesar 當然不會直接傳 “APPLE” (如果內容被其他人看到，機密不就外洩了嗎！！) 於是生性多疑的 Caesar 會根據 **ALPHABET** 每個字母所對應到的 **new_alphabet** 字母位置寫下加密文字 “WLLHA”，詳細字母的對應方式如下圖所示：

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
W X Y Z A B C D E F G H I J K L M N O P Q R S T U V

(A 字母對應到新字串的 W // P 對應到 L // L 對應到 H // E 對應到 A)

收到“WLLHA”的 Jerry 又該怎麼知道 Caesar 是在說什麼呢？原來 Caesar 傳來的信封裡，除了寫著“WLLHA”的白紙之外，抖一抖，裡面又藏著一張寫著“4”的小紙條，告訴 Jerry 請把我們所熟知的英文字母串 **ALPHABET** 向右平移 4 格、得到 **new_alphabet**、再將“WLLHA”的每個字母對應回沒有平移的英文字母串 **alphabet**，就可以得到“APPLE”了！解密的過程如下圖所示：



請同學完成名為 `caesar.py` 的檔案，並完成解密（`decipher`）的過程。當您完成時，程式可以完美重現下圖中每一行文字與數字：

```
Secret number: 4
What's the ciphered string?WLLHA
The deciphered string is: APPLE
```

當同學順利解出一個單字後，請思考一下：如果今天傳來的機密是一個完整的句子，我們該怎麼解呢？（您的程式最終應該要能完美重現下圖的每一行文字與數字）

```
Secret number: 7
What's the ciphered string?RHN TKX MAX UXLM!
The deciphered string is: YOU ARE THE BEST!
```

注意：您的程式應該是 **case-insensitive**。也就是使用者輸入的文字大小寫不會影響最後得到的答案（如下圖所示）

```
Secret number: 7
What's the ciphered string?rhn TKx Max UXLM!
The deciphered string is: YOU ARE THE BEST!
```


Problem 5 - hangman.py

最後一題將請同學使用 `Console` 完成經典猜字遊戲 `Hangman` (吊死鬼)



程式的一開始會從字庫隨機選擇一個英文單字 (以下簡稱為 **answer**)，並將每一個字母用橫槓遮住 (以下簡稱為 **dashed**)。玩家每次輸入一個大寫或寫小的字母 (以下簡稱為 **input_ch**)，如果 **input_ch** 存在於 **answer** 之中，程式就會更新 **dashed** 並把所有 **input_ch** 所在的位置展示出來。然而，如果 **input_ch** 不存在於 **answer** 之中，玩家就會損失一條命。若七條命扣完還沒猜出來，玩家挑戰失敗

真正的 `Hangman` 遊戲會在玩家猜錯時更新吊死鬼的圖樣 (如下圖所示)，當吊死鬼的頭部、身體、左手、右手、左腳、右腳、舌頭都被呈現出來時玩家挑戰失敗



然而，我們在這題並不需要做到圖樣的部分！只要完成 `Console` 版的即可 (或許圖樣版的可以當成各位的 `Extensions` ^^)

您的程式將可以完美呈現下圖的每一行文字與結果：

```

The word looks like: -----
You have 7 guesses left.
Your guess: k
There is no K's in the word.
The word looks like: -----
You have 6 guesses left.
Your guess: k
There is no K's in the word.
The word looks like: -----
You have 5 guesses left.
Your guess: j
There is no J's in the word.
The word looks like: -----
You have 4 guesses left.
Your guess: q
There is no Q's in the word.
The word looks like: -----
You have 3 guesses left.
Your guess: x
There is no X's in the word.
The word looks like: -----
You have 2 guesses left.
Your guess: b
There is no B's in the word.
The word looks like: -----
You have 1 guesses left.
Your guess: a
There is no A's in the word.
You are completely hung : (
The word was: REFUND

```

```

The word looks like: -----
You have 7 guesses left.
Your guess: I
You are correct!
The word looks like: ----I---I--
You have 7 guesses left.
Your guess: i
You are correct!
The word looks like: ----I---I--
You have 7 guesses left.
Your guess: h
You are correct!
The word looks like: H---I---I--
You have 7 guesses left.
Your guess: o
You are correct!
The word looks like: H0---I---I--
You have 7 guesses left.
Your guess: P
You are correct!
The word looks like: H0-PI---I--
You have 7 guesses left.
Your guess: t
You are correct!
The word looks like: H0-PIT---IT-
You have 7 guesses left.
Your guess: a
You are correct!
The word looks like: H0-PITA-IT-
You have 7 guesses left.
Your guess: L
You are correct!
The word looks like: H0-PITALIT-
You have 7 guesses left.
Your guess: s
You are correct!
The word looks like: HOSPITALIT-
You have 7 guesses left.
Your guess: Y
You are correct!
You win!!
The word was: HOSPITALITY

```

以下五點重點提醒

1. 請使用我們寫好的 `random_word()` 來得到一個隨機的英文單字。同學不用了解此 `function` 的每一行 `code`，只要知道 `random_word()` 每一次會隨機 `return` 一個英文單字出來即可

2. 使用者的輸入為 **case-insensitive**，**upper case (大寫)** **lower case (小寫)** 都可以
3. 輸入兩次錯誤答案還是會少一條命（如左圖 **k** and **k**）
4. 輸入兩次正確答案還是得到一樣的結果（如右圖 **l** and **i**）
5. 當使用者輸入格式錯誤，例如輸入的不是英文字母(可以用 **str.isalpha()** 判斷)或不只一個字母，程式應該要印出 **"Illegal format."** 並重複要求使用者輸入，直到格式正確（如下圖所示）

```
The word looks like: -----
You have 7 guesses left.
Your guess: 2
illegal format.
Your guess: aa
illegal format.
Your guess: e
You are correct!
The word looks like: -E-----
You have 7 guesses left.
Your guess:
```

評分標準

Functionality - 程式是否有通過我們的基本要求？程式必須沒有 **bug** 、能順利完成指定的任務、並確保程式沒有卡在任何的無限迴圈（**infinite loop**）之中。

Style - 如同我們在課堂上所說，好的程式要有好的使用說明，也要讓人一目瞭然，這樣全世界的人才能使用各位的 **code** 去建造更多更巨大更有趣的程式。因此請大家寫**精簡扼要**的 **main()**程式概要、**function comments** 和單行註解。

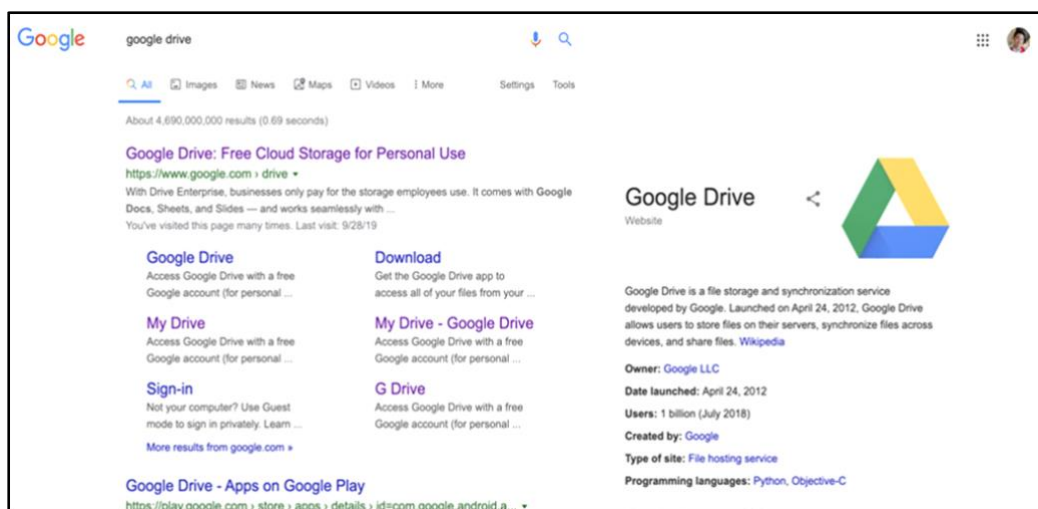
作業繳交

恭喜您完成 **Assignment 3**！大家應該要對自己的成就感到驕傲，因為這份作業跟史丹佛大學的學生作業非常相似，代表您跟世界各國的菁英一樣厲害了。

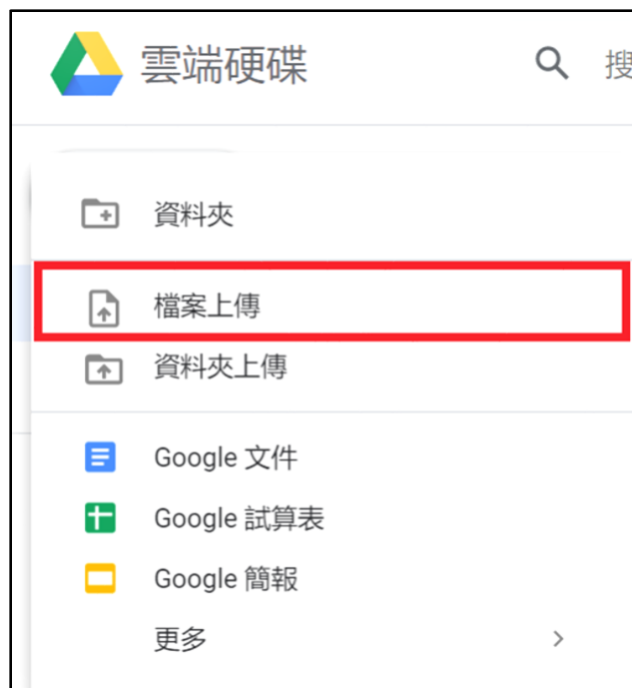
請同學於**作業繳交期限前**，依照下圖將您完成的作業的**下載連結**上傳至社團提供的**作業繳交表單**。



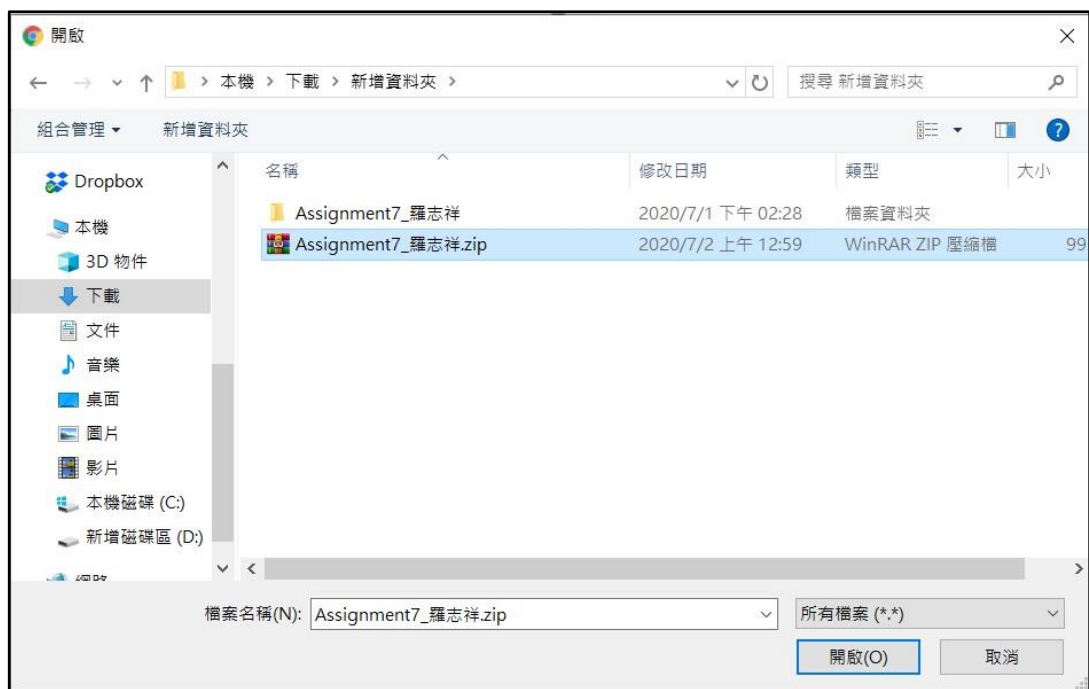
找到作業資料夾，按右鍵



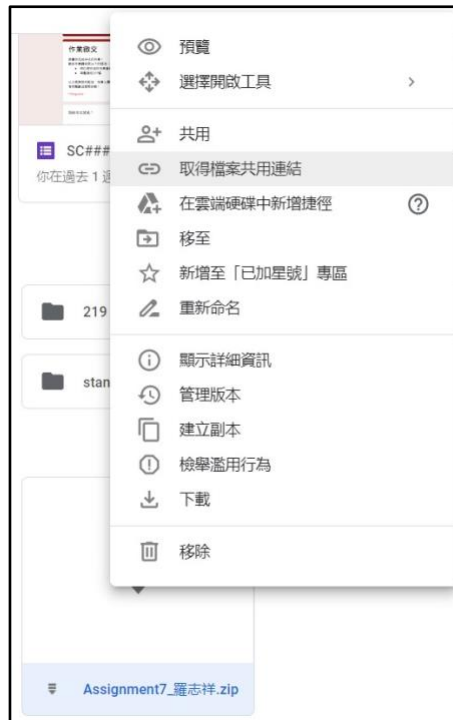
上網搜尋 Google Drive



點選 檔案上傳(File Upload)



找到剛剛壓縮的 **Assignment3** 檔案後，點選 開啟(Open)



上傳完成後，對檔案按右鍵，選擇 取得檔案共用連結(Get Shareable Link)



將「限制(Restricted)」改成「知道連結的使用者(Anyone With the Link)」

再按下 複製連結

點開社團提供的作業繳交表單：

stanCode

作業繳交

恭喜您完成本次的作業！

繳交作業請依照以下的規定：

- 將已經完成的作業資料夾命名成以下格式「Assignment#_Name」
- 再壓縮成ZIP檔

作業以最後繳交的版本計算成績。
若有提交新版本，請通知各自的助教，以免遺漏。
以上，謝謝配合～

*必填

您的中文姓名：*

您的回答

您的作業連結：（請確定已經開啟權限）*

您的回答

提交

填入相關資訊，並在最後附上剛剛複製的連結

確認無誤後按下提交

stanCode - 標準程式教育機構

stanCode