

SC201 Assignment0



歡迎來到 SC201！ stanCode的AI班將會用到大量的Python資料結構、檔案處理、文字處理，以及部分優化演算法。因此，本份作業的目的為提供同學前述重點的總複習，我們也會透過這份作業更了解您對這些程式概念是否熟悉。

本分檔案的所有題目皆改編自我們的進階班作業與考試，請您於**開課前**，依據下方作答指示完成此份檔案的題目，如有任何問題，歡迎來信 stancode.tw@gmail.com 詢問。

1. 新增一份自己的 Google Doc 空白文件
2. 打開 [作答模板](#)，全選（Ctrl(Command) + A）整份文件內容
3. 按下（Ctrl(Command) + C）複製
4. 將複製的內容貼上自己的 Google Doc 空白文件，開始作答

作業繳交期限：請於**開課前**依照下方作業繳交指示，上傳至課程社團中作業貼文提供的「作業提交表單」。

Problem 1: Recursion

請追蹤下圖程式碼，寫出 `print(mystery(348))` 會在 Console 上印出什麼？
(請寫出追蹤過程。只有答案此題無法獲得分數)

完成後，請將您的追蹤過程拍照，並貼到 Google doc 作答模板上。

```
def mystery(n):  
    if n < 10:  
        return 10*n+n  
    else:  
        a = mystery(n // 10)  
        b = mystery(n % 10)  
        return 100*a+b
```

Problem 2: Code Trace

請問下圖程式碼在 Console 印出的四行文字與數字為何？請符合格式，將所有執行此程式後會看到的一切文字與數字寫在作答模板中：

```
def reference_mystery():
    a = [10, 20]
    b = 3
    c = 5

    mystery(a,b,c)
    print(str(a), b, c)

    a[1] += 1
    mystery(a, a[0], b)
    print(str(a), b, c)

def mystery(a, b, c):
    b = b + c
    for i in range(len(a)):
        a[i] += 1
    c = c + a[0]
    print(str(a), b, c)

if __name__ == '__main__':
    reference_mystery()
```

Problem 3: Nested Data Structure

在美國 Social Security Administration 網站有提供一份資料：每一年女性、男性新生兒前五名的人氣排名！如下圖：

2019, Olivia, Emma, Ava, Sophia, Isabella, Liam, Noah, Oliver, William, Elijah
2018, Emma, Olivia, Ava, Isabella, Sophia, Liam, Noah, William, James, Oliver
2017, Emma, Olivia, Ava, Isabella, Sophia, Liam, Noah, William, James, Logan
2016, Emma, Olivia, Ava, Sophia, Isabella, Noah, Liam, William, Mason, James
2015, Emma, Olivia, Sophia, Ava, Isabella, Noah, Liam, Mason, Jacob, William
2014, Emma, Olivia, Sophia, Isabella, Ava, Noah, Liam, Mason, Jacob, William
2013, Sophia, Emma, Olivia, Isabella, Ava, Noah, Jacob, Liam, Mason, William
2012, Sophia, Emma, Isabella, Olivia, Ava, Jacob, Mason, Ethan, Noah, William

stanCode 需要各位同學處理該文字檔並計算每個名字的「分數」。而「分數」怎麼算呢？我們給予獲得第一名的名字 5 分、第二名 4 分、第三名 3 分... 以此類推

這個文字檔 `popularity.csv` 每一列開頭都是年份；年份後緊接著女性姓名的前五名；隨後是男性姓名的前五名。以 2019 資料為例，女性姓名前五名為「Olivia, Emma, Ava, Sophia, Isabella」；男性姓名前五名為「Liam', 'Noah', 'Oliver', 'William', 'Elijah」

請在第 5, 6 頁編輯名為 `get_dict()` 的函式，讀取 `popularity.csv` 之文字檔並將它整理成如下所示之 Python dictionary (以下簡稱 `all_d`)

```
{  
    'female': {'Olivia': 30, 'Emma': 37, 'Ava': 17, 'Sophia': 22, 'Isabella': 14},  
    'male': {'Liam': 30, 'Noah': 34, 'Oliver': 4, 'William': 15, 'Elijah': 1, ...}  
}
```

`all_d` 是一個「以性別為 key、以 `name_score_d` 為 value」的 Python dictionary。
`name_score_d` 是一個「以姓名為 key、以分數為 value」的 Python dictionary。

以下 3 點請注意：

1. 每一行文字結尾都有換行字元 '\n'。存取前記得把它去除
2. 您的程式應該要可以處理所有符合這個格式的 csv 檔! 也就是說，年份可以不只到 2012、且若文字檔內儲存女生/男生的前六名照樣可以用相同程式碼計算出來! (第一名獲得的分數變成 6 分、第二名 5 分、... 以此類推)
3. 女性排名一定會在男性前面，且名次數目會一樣（前五名就男女都是前五名、前六名就男女都是前六名）

請將答案寫在作答模板中。

FILENAME = 'popularity.csv'

This file contains name ranks each year

def get_dict():

"""

: return: all_d (dict{int: dict{str: int}}): Dictionary holding sex as key,
name_score_d as value.

(name_score_d holds name as key, rank as value)

"""

Your Code Here

Problem 4: Class Design - Facebook, Google 軟體工程師面試題

stanCode 希望同學們幫忙設計一個可以快速搜尋英文單字的資料結構！

Lexicon (英文意義為「詞庫」) 的 class constructor 不需接收任何資訊。由 **Lexicon()** 製造出來的 objects 支援**加入單字** (以下簡稱 **add_word**)、**搜尋單字** (以下簡稱 **search**) 這 2 項功能。假設我們要製造一個 **Lexicon** 資料結構的 object，輸入的指令如下方所示：

```
lexicon = Lexicon()
```

Lexicon 這個 class 裡包含了 1 個 constructor 與 2 個 instance methods：

```
def __init__(self):
    """
    Initialize your data structure here.
    """

def add_word(self, word):
    """
    : param word: str, the word that needs to be stored into your data structure.
    -----
    Add the word into your data structure. The word will never be a duplicate.
    """

def search(self, word):
    """
    : param word: str, the word to be checked if it is in your data structure.
    : return: bool, if the word is in your data structure or not.
    -----
    Search through your data structure to see if the word is in or not
    """
```

為了讓同學更清楚了解每個 instance methods, 我們以上方製造的 **lexicon** 來舉例：

- **lexicon.add_word('apple')** 會把 'apple' 加入您的資料結構
- **lexicon.search('apple')** 會 return **True**
- **lexicon.search('app')** 會 return **False**

然而，**search** 支援輸入以 . 取代英文字母的功能。舉例來說，若接續上述例子：

- **lexicon.search('app..')** 會 return **True**
- **lexicon.search("..p.e")** 會 return **True**
- **lexicon.search('.....')** 會 return **True**
- **lexicon.search('...')** 會 return **False**

您可以確保每次加入的 **word** 長度一定大於等於 1。除此之外，您也可以假設 **word** 裡面只包含小寫英文字母。

請將答案寫在作答模板中。

***class* Lexicon:**

Your Code Here

作業繳交

恭喜各位同學完成 SC201 Assignment0! 這份作業將帶領同學前往更艱深的觀念 & AI 世界。這份作業的難度與史丹佛大學學生作業非常相似，代表各位同學也成為世界上最強的一群了！

請同學依循下圖，完成作業上傳。

1. 開啟連結共用設定，並複製下載連結

- 1) 對檔案按右鍵，點選「共用」
- 2) 點擊「變更任何知道這個連結的使用者權限」後，權限會變為「可檢視」
- 3) 點選「複製連結」



2. 待加入課程臉書社團後，將連結上傳至作業貼文提供的「作業提交表單」