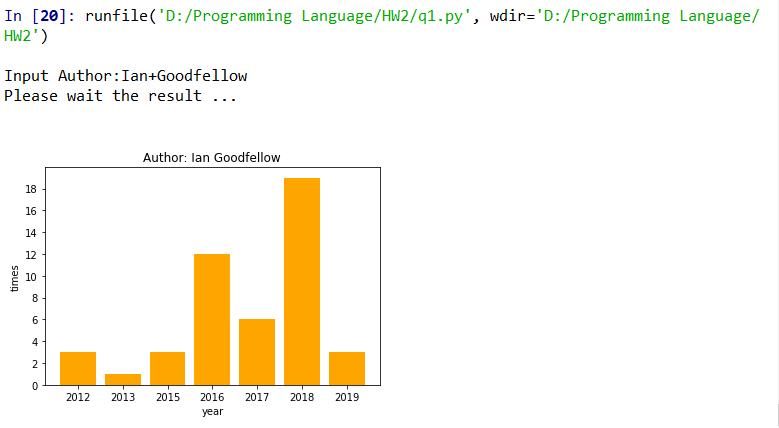
**執行環境：皆在Windows 10的Spyder裡執行(Python3.7)**

**Question 1**

執行q1.py後，需輸入作者的名字，而後會出現Please wait the result ...，等待結果爬完後，即會出現bar chart。

範例Output為下圖：

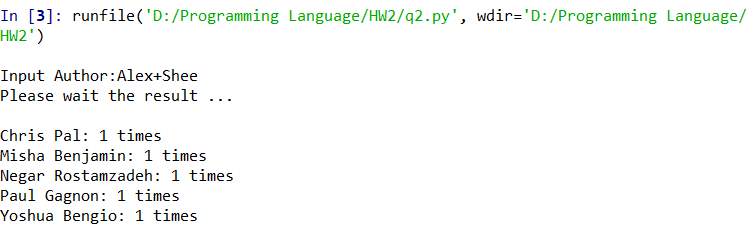


**程式碼說明:**

在使用者輸入作者名字後，便會開始以作者姓名開始進行爬蟲，而網址預設一頁的顯示結果數量為200(減少換頁次數，加速輸出速度)，若遇到有下一頁的情況，則會繼續前往下一頁抓取資料(是否有下一頁以next\_page這個參數來判斷，假如next\_page為None則表示已無下一頁)。而搜尋到的每篇論文都會檢查author欄位裡面是否有和input相同的author(以author\_exist這個參數來判斷，None表示此篇論文的author和input的author不相同，因此不列入計算)，若author\_exist不為None，才獲取originally announced的年份後，將其記錄在Record這個dictionary裡面(key為年分，Value為次數)，若該年分已存在此dictionary裡，則將他的value+1，若不存在，將其年份加入並把value設為1。把各個paper的年份都記錄完後，按照年分進行Sort(讓bar chart依照年分繪出)，而後利用matplotlib將Record的資料用bar chart畫出。

**Question 2**

執行q2.py後，需輸入作者的名字，而後會出現Please wait the result ...，等待結果爬完後，即會印出co-author的結果。

範例Output為下圖：

**程式碼說明:**

在使用者輸入作者名字後，便會開始以作者姓名開始進行爬蟲，而網址預設一頁的顯示結果數量為200(減少換頁次數，加速輸出速度)，若遇到有下一頁的情況，則會繼續前往下一頁抓取資料(是否有下一頁以next\_page這個參數來判斷，假如next\_page為None則表示已無下一頁)，而搜尋到的每篇論文都會檢查author欄位裡面是否有和input相同的author(以author\_exist這個參數來判斷，None表示此篇論文的author和input的author不相同，因此不列入計算)，若author\_exist不為None，才將該篇論文的co-author資訊記錄在Record這個dictionary裡面(key為co-author姓名，Value為次數)，若該姓名已存在此dictionary裡，則將他的value+1，若不存在，將其姓名加入並把value設為1。把各個paper的co-author姓名都記錄完後，按照姓名進行Sort，再將co-author姓名(key)以及相對應的次數(value)印出來(不包含一開始input的作者)。