執行環境：mac terminal (using vscode)

執行步驟：python Main.py 名稱(直接打名稱即可，不用去空白)

程式碼說明：

- 執行時是用 argv，先把名稱先組起來 -- @a
- 再搜尋對應網址的 html -- @b
- 判斷是否有沒有下一頁，找是否有無"pagination-list" -- @c
- 如果有，則先取出"下一頁的網址" -- @d
- 如果沒有，也把原本網址存在跟"下一頁的網址"一樣的變數內 -- @e
- 之後用逐個網址去取出 html -- @f
- 因為搜尋名字可能會搜尋出相似的作者但不是正確的作者名，所以必須先取出所有 co-worker，確認是否有要的名字在裡面，才能繼續下去 -- @h
- 取出所有的年份之後，再利用 dictionary 去存進 data_Problem_1 中 -- @i
- 取出所有的作者之後，再利用 dictionary 去存進 data_Problem_2 中 -- @j
- 輸出 problem 2，利用 sort 去排順序，並把搜尋作者過濾掉 -- @k
- 輸出 problem 1 柱狀圖，利用 sort 去排順序 -- @l

```
import matplotlib.pyplot as plt
import sys
import re
import urllib.request

##### get particular html #####
original_Author = sys.argv[1]      ## @a
author = sys.argv[1]
if len(sys.argv) is not 2:
    for inputAuthor in sys.argv[2:]:
        author = author + "+" + inputAuthor
        original_Author = original_Author + " " + inputAuthor
url = "https://arxiv.org/search/?query=" + author +
"&searchtype=author&abstracts=show&order=-announced_date_first&size=50"
content = urllib.request.urlopen(url)   ## @b
html_str = content.read().decode("utf-8") # get all html
```

```
##### if have next page #####
is_Next_Page = False
target_Url = []
try:     ## @c
    is_Next_Page = True
    nextPage_Pattarn = "pagination-list[\s\S]*?</ul>"
    nextPage_Result = re.findall(nextPage_Pattarn, html_str)
except:
    is_Next_Page = False

if is_Next_Page is True: ## @d
    ## get url
    tmp_Result = nextPage_Result[0].split('pagination-list">')[1].split("</a>")[0:-1] #[0:-1]
get rid of the last useless data
    for tmp in tmp_Result:
        tmp_Url = tmp.split('<a href=')[1].split("class")[0].strip()[1:-1]
        target_Url.append("https://arxiv.org" + tmp_Url.replace("amp;", ''))
else:
    target_Url.append(url)     ##@e




data_Problem_1 = {}
data_Problem_2 = {}
print("[ Author: " + author + " ]")
for tmp_Url in target_Url:     ## @f
    ## get next page url
    if is_Next_Page is True:
        content = urllib.request.urlopen(tmp_Url)
        html_str = content.read().decode("utf-8")  # get all html

    pattarn = 'Authors:</span>[\s\S]*?</li>'  # get Name of author
    result = re.findall(pattarn, html_str)       ## @h
```

```
    for r1 in result:
        name = r1.split('</p>')[0]
        name = name.split('</a>')[:-1]

        ## check if the source is right author
        tmp_Name_List = []     ## @h get name list
        for n in name:
            tmp_Name = n.split('">')[1].strip()
            tmp_Name_List.append(tmp_Name)

        ## @h if not then continue
        if not(original_Author in tmp_Name_List): continue # if not the right author

        ## get problem 1 data
        ## @i
        pattarn1 = "originally announced</span>[\s\S]*?</p>"  # get year
        result1 = re.findall(pattarn1, r1)
        for r2 in result1:
            ## @i 取出年份
            year = r2.split('</span>')[1].split(".")[0].strip().split(" ")[1].strip()

             ## @i 將資料加進 dictionary
            if year in data_Problem_1:
                data_Problem_1[year] = data_Problem_1[year] + 1
            else:
                data_Problem_1[year] = 1


        ## get problem 2 data
         ## @j 因為前面要確認名字，因此沿用前面抓出的 name list 即可，將資料加進
dictionary
        for nn in tmp_Name_List:
            if nn in data_Problem_2:
                data_Problem_2[nn] = data_Problem_2[nn] + 1
```

```
            else:
                data_Problem_2[nn] = 1


## Print Problem 2
## @k
for data in sorted(data_Problem_2):
    if not(data == original_Author):    ## @k 過濾掉搜尋的作者
        print("[" + data + "]: " + str(data_Problem_2[data]) + " times")


## Print Problem 1
## @l  x 軸為 key，y 軸為 value
plt.bar(sorted(data_Problem_1.keys()), data_Problem_1.values())
plt.show()
```