第1次作業-作業-HW1

> 學號：1121113234
> 姓名：阮陳家興
> 作業撰寫時間：180 (mins，包含程式撰寫時間)
> 最後撰寫文件日期：2023/09/22

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- ☑ 說明內容
- ☑ 個人認為完成作業須具備觀念

說明程式與內容

開始寫說明，該說明需說明想法， 並於之後再對上述想法的每一部分將程式進一步進行展現， 若需引用程式區則使用下面方法， 若為.cs檔內程式除了於敘述中需註明檔案名稱外， 還需使用語法```語言種類 程式碼 ```，其中語言種類若是要用python則使用py，java則使用java，C/C++則使用cpp， 下段程式碼為語言種類選擇csharp使用後結果：

```
public void mt_getResult(){
    ...
}
```

若要於內文中標示部分網頁檔，則使用以下標籤```html 程式碼 ```， 下段程式碼則為使用後結果：

```
<%@ Page Language="C#" AutoEventWireup="true" ...>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" ...>
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
        </div>
    </form>
</body>
</html>
```

更多markdown方法可參閱https://ithelp.ithome.com.tw/articles/10203758

請在撰寫"說明程式與內容"該塊內容，請把原該塊內上述敘述刪除，該塊上述內容只是用來指引該怎麼撰寫內容。

1. 請解釋何謂git中下列指令代表什麼？並舉個例子，同時必須說明該例子的結果。其指令有add、commit、push、fetch、pull、branch、checkout與merge。

Ans:

- git add
  git add 指令可用於將檔案新增至索引。例如，下列命令可以將本機目錄中名為 temp.txt 的檔案新增至索引：

```
git add temp.txt
```

- git commit
  git commit 指令用於從頭開始提交更改。

```
git commit
```

- git push
  git push 是一個廣泛使用的 git 指令。一個簡單的推送命令會將變更傳送到使用目錄指定的遠端伺服器儲存庫的主分支。例如：

```
git push
```

- git fetch
  git fetch 允許使用者從遠端儲存庫下載本機目錄中不存在的所有物件。使用範例：

```
git fetch origin
```

- git pull
  要將遠端儲存庫上的所有變更合併到本機運作目錄，需要使用 pull 指令。使用：

```
git pull
```

- git branch
  在 Git 儲存庫中，總是有許多單獨的分支，用於獨立於其他分支部署某個功能，分支命令 用於顯示所有現有分支。

```
git branch
```

- git merge
  git merge 指令用於將分支合併到活動分支。使用：

```
git merge <branch-name>
```

- git checkout
  git checkout 指令可用於建立分支或在分支之間切換。例如，以下命令建立一個新分支並跳到它；

```
command git checkout -b <branch-name>
```

若要從一個分支切換到另一個分支，請使用：

```
git checkout <branch-name>
```

2. 於專案下的檔案—**hw1.py**，撰寫註解，以說明該程式每列中之背後意義。

   該hw1.py題目如下：

```
統計字母數。假設今天輸入一句子，句子中有許多單字，單字皆為英文字母小寫，
請統計句子中字母出現的字數，輸出實需要照字母排序輸出，且若該字母為0則不輸出

如輸入
this is an apple
輸出
a: 2
e: 1
h: 1
i: 2
l: 1
n: 1
p: 2
s: 2
t: 1
```

Ans:

```
# 此指令用於從類型庫匯入List資料類型，以確定變數和參數的類型
from typing import List

#定義了一個名為 countLetters 的函數，countLetters 函數接受單一參數、句子，並使用資料型態 str 進行註釋
def countLetters(sentence: str) -> List[int]:
```

```
    #letterCount：List[int]：是 letterCount 變數的資料型別註解。指示 letterCount 是一個包含整數值的清單。用 26 個元素初始化列表，每個元素的值為 0
    letterCount: List[int] = [0] * 26

    #此命令列有助於對字串「sentence」的每個字元執行特定操作（在循環內定義）。
    for char in sentence:

        #檢查字串中的目前字元是否為字母。如果該字元是字母，則將執行 if 區塊中的下一行程式碼。如果字元不是字母，則 if 內的語句區塊將被忽略。
        if char.isalpha():

            #此命令列將 char 字元轉換為表示其在字母表中的位置的數字（從 0 開始）。
            index = ord(char) - ord('a')

            #命令列用於統計字串中每個字母出現的次數。
            letterCount[index] += 1

    #該命令將傳回一個列表，其中包含輸入字串中每個字母的出現次數。
    return letterCount

#當您還不想編寫程式碼或想跳過程式的特定部分而不引起錯誤時，可以使用 pass 關鍵字。
pass

#此函數接受表示字母出現次數的整數列表並列印該資訊。它不會傳回任何值，而只是在螢幕上執行列印操作。
def printLetterCount(letterCount: List[int]) -> None:

    #命令列設定了26次循環，i的值分別為0到25。
    for i in range(26):

        #檢查 letterCount 清單中位置 i 處字母的出現次數是否大於 0
        if letterCount[i] > 0:

            #命令列用於列印letterCount清單中每個字母出現的次數
            print(f"{chr(i + ord('a'))}: {letterCount[i]}")
pass

#此命令列會建立一個包含字串「this is an apple」的 inputSentence 變數
inputSentence: str = "this is an apple"

#命令列使用 inputSentence 字串呼叫 countLetters 函數，接收傳回值作為字串中字母出現次數的列表，並將其指派給 letterCount 變數以供進一步使用。
letterCount: List[int] = countLetters(inputSentence)

#命令列列印字串中每個字母出現的次數。
printLetterCount(letterCount)
```

3. 請新增檔案**hw1_2.py，**輸入一個正整數(N)，其中$1\le N \le 100000$，請將該正整數輸出進行反轉

```
如輸入
1081
```

```
輸出
1801

如輸入
1000

輸出
1
```

Ans:

```
# # 讀輸入整數類型。
N: int=int(input("Please input an integer (1 ≤ N ≤ 100000): "))

# 把整數轉到字符串並反轉，轉回整數。
R: int=int(str(N)[::-1])

# 輸出結果。
print(R)
```

4. **[課外題]**：請找尋資料，說明何謂**單元測試**，請新增檔案**hw1_3.py**，並利用溫度計攝氏轉華氏撰寫單元測試。

Ans:

```
# test_hw1_3.py
import unittest

def convert_temperature(celsius: float) -> float:
 """將攝氏溫度轉換為華氏溫度"""
 return (celsius * 9/5) + 32

class TestTemperatureConversion(unittest.TestCase):

 def test_freezing_point(self):
    self.assertEqual(convert_temperature(0), 32, "冰點轉換失敗")

 def test_boiling_point(self):
    self.assertEqual(convert_temperature(100), 212, "沸點轉換失敗")

 def test_negative_temperature(self):
    self.assertEqual(convert_temperature(-40), -40, "負溫度轉換失敗")

 def test_fractional_temperature(self):
    self.assertAlmostEqual(convert_temperature(36.6), 97.88, places=2, msg="體溫轉
換失敗")

if __name__ == '__main__':
 unittest.main()
```

個人認為完成作業須具備觀念

開始寫說明，需要說明本次練習需學會那些觀念 (需寫成文章，需最少50字，並且文內不得有你、我、他三種文字)且必須提供完整與練習相關過程的notion筆記連結