

第1次作業-作業-HW1

學號：112111204

姓名：邱元平

作業撰寫時間：??? (mins · 包含程式撰寫時間)

最後撰寫文件日期：2025/01/06

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: add(新增) 範例：git add[<檔案名稱.副檔名>或.><-all>新增全部] 結果：將檔案加入索引 commit(提交) 範例：git commit 結果：將檔案提交上成為一個新版本 push(推送) 範例：git push 結果：想當然提交完版本後就是上傳·將版本從本地推送至GitHub上 fetch(獲取) 範例：git fetch 結果：從GitHub上下载所有新的資料 pull(拉取) 範例：git pull 結果：將伺服器端最新的倉使分支(branch)與客戶端的進行對應分支合併(merge) branch(分支) 範例：git branch(分支名稱) 結果：割建新分支 checkout(切换) 範例：git checkout(分支名稱) 結果：切换至所選分支 merge(合併) 範例：git merge(分支名稱) 結果：將所選分支名稱併入當下所處分支

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: from typing import List #導入Typing函式庫中的List

```
def countLetters(sentence: str) -> List[int]: #定義一個計算字母出現次數的自訂函數
letterCount: List[int] = [0] * 26 #創立一個長度26的列表，用來儲存每個字母的次數 [a-z]
```

```
for char in sentence: #利用For迴圈遍布所有字母
    if char.isalpha(): #如果字元為字母則執行以下
        index = ord(char) - ord('a') #判斷字母所處的順序位置
        letterCount[index] += 1 #將對應字母次數增加1

return letterCount #返回此自訂函數列表
```

pass

```
def printLetterCount(letterCount: List[int]) -> None: #定義一個列印字母出現次數的自訂函數
for i in range(26):
    if letterCount[i] > 0: #如果這個字母出現次數大於0，執行以下
        print(f"{chr(i + ord('a'))}: {letterCount[i]}") #列印出該字母和出現次數
pass
```

```
inputSentence: str = "this is an apple" #輸入所要檢測的字串 letterCount: List[int] =  
countLetters(inputSentence) #計算字母出現次數 printLetterCount(letterCount) #列印字母出現次數
```

3. 請新增檔案**hw1_2.py**，**輸入一個正整數(N)，其中 $1 \leq N \leq 100000$ ，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: while True: n = int(input("請輸入一個正整數 (1 <= N <= 100000): ")) if 1 <= n <= 100000: print(int(str(n)[::-1])) break #Break作用 => 如果輸入正確則會執行，執行後跳出循環 else: print("輸入的數字超出範圍，請重新輸入！")

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:

主程式 攝氏轉華氏的函數

```
def celsius_to_fahrenheit(celsius: float) -> float: return celsius * 9 / 5 + 32
```

單元測試 使用 unittest 模組進行測試

```
import unittest
```

```
class TestTemperatureConversion(unittest.TestCase): # 測試正數攝氏轉換 def test_positive_celsius(self):  
self.assertAlmostEqual(celsius_to_fahrenheit(100), 212) # 100°C = 212°F  
self.assertAlmostEqual(celsius_to_fahrenheit(0), 32) # 0°C = 32°F  
self.assertAlmostEqual(celsius_to_fahrenheit(37), 98.6) # 37°C ≈ 98.6°F
```

```
# 測試負數攝氏轉換
```

```
def test_negative_celsius(self):
```

```
    self.assertAlmostEqual(celsius_to_fahrenheit(-40), -40) # -40°C = -40°F
```

```
    self.assertAlmostEqual(celsius_to_fahrenheit(-20), -4) # -20°C = -4°F
```

```
# 測試邊界值轉換
```

```
def test_boundary_celsius(self):
```

```
self.assertAlmostEqual(celsius_to_fahrenheit(1), 33.8)    # 1°C ≈ 33.8°F  
self.assertAlmostEqual(celsius_to_fahrenheit(-1), 30.2)   # -1°C ≈ 30.2°F
```

if **name** == '**main**': unittest.main() #單元測試是針對程式來進行檢驗的測試工作。

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