第1次作業-作業-HW1

> 學號：112112105
> 姓名：李佩琪
> 作業撰寫時間：100 (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:

```
add ( 加入 )

例子：創建一個名為 ty 的檔案，然後在終端中輸入 git add ty
結果：這個命令會把檔案 ty 加入到暫存區，以準備提交。
commit ( 提交 )

例子：在終端中輸入 git commit
結果：這會將暫存區中的 ty 檔案提交到本地的 Git 倉庫。
push ( 推送 )

例子：在終端中輸入 git push
結果：這個操作會把已提交的更改從本地倉庫推送到遠端伺服器。
fetch ( 獲取 )

例子：在終端中輸入 git fetch
結果：此命令會從遠端倉庫獲取最新的資料，但不會合併到本地分支。
pull ( 拉取 )

例子：在終端中輸入 git pull
結果：這會從伺服器獲取新的資料並將其合併到本地分支，相當於執行 fetch 和 merge 的結合。
branch ( 分支 )

例子：在終端中輸入 git branch new
結果：此命令會創建一個新的分支，名為 new。
checkout ( 切換 )

例子：在終端中輸入 git checkout new
結果：這會將當前工作目錄切換到 new 分支。
merge ( 合併 )

例子：在終端中輸入 git merge new
結果：這個操作會把 new 分支的變更合併到當前分支中。
```

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

def countLetters(sentence: str) -> List[int]:
    # 初始化一個列表，長度為26，分別計數a-z的出現次數
    letterCount: List[int] = [0] * 26

    # 遍歷句子中的每個字符
    for char in sentence:
        if char.isalpha():  # 判斷字符是否為字母
            index = ord(char.lower()) - ord('a')  # 獲取字母的索引（將字母轉為小寫）
            letterCount[index] += 1  # 將此字母計數+1

    return letterCount  # 返回每個字母的計數列表

def printLetterCount(letterCount: List[int]) -> None:
    # 遍歷26個字母
    for i in range(26):
        if letterCount[i] > 0:  # 如果對應字母的計數大於0，表示該字母出現過
            print(f"{chr(i + ord('a'))}: {letterCount[i]}")  # 顯示字母及其出現次數

# 輸入字符串，計算字母出現次數
inputSentence: str = "this is an apple"  # 將句子設為 "this is an apple"
letterCount: List[int] = countLetters(inputSentence)  # 計算各字母出現的次數
printLetterCount(letterCount)  # 輸出每個字母的計數
```

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:

```
# 輸入一個正整數
num = input("請輸入一個正整數: ")

# 將此數字進行反轉
reversed_num = num[::-1]

# 去掉前導零
reversed_num = int(reversed_num)

# 輸出反轉後的數字
print("反轉後的數字是:", reversed_num)
```

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:

單元測試（Unit Testing）是一種軟體測試方法，旨在驗證軟體中的最小可測單元（通常是函數或方法）是否按預期工作。以下是單元測試的幾個關鍵特點：

最小測試單位：單元測試專注於單一的程式碼單元，通常是函數或方法，並測試其輸入和輸出。

自動化：單元測試通常會自動執行，這使得測試的執行變得迅速和高效。許多程式設計語言和框架都提供了支援單元測試的工具。

及早發現錯誤：通過在開發過程的早期階段進行單元測試，開發人員可以及早發現並修正錯誤，減少後續測試階段的負擔。

促進重構：單元測試可以為程式碼重構提供信心。當開發人員改動現有代碼時，運行單元測試可以確認改動不會破壞原有功能。

提高可維護性：良好的單元測試可以幫助其他開發人員理解程式碼的功能，增強團隊間的協作。

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

本次練習旨在掌握字串處理及數據反轉的基本概念。透過對正整數進行反轉，學習如何使用字串切片技術進行操作，並能有效地去除前導零，確保輸出結果的正確性。此外，對使用者輸入的處理與檢查，增強了對邊界條件的理解。整個過程包括接收輸入、字串反轉、數據類型轉換及輸出結果，提升了對 Python 語言的操作熟練度。此練習為後續更複雜的數據處理打下基礎。