

第2次練習-練習-PC2

學號：112111234

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. 請參照Topic2投影片p. 39, ...(見題目.pdf)

Ans:

```
##第1題
# 定義多項式  $f(x) = 6x^4 + 2x^2 + 3$ 
def f(x):
    return 6 * (x**4) + 2 * (x**2) + 3

# 儲存結果的字典格式
polynomial = {
    "coefficients": [4, 6, 0, 2, 3, 0], # 係數對應  $[6x^4, 0x^3, 2x^2, 0x, 3]$ 
    "degree": 4 # 多項式的最高次數
}

# 計算  $x = 91$  時的值
x = 91
result = f(x)

# 顯示結果
print()
print(f"f({x}) = {result}")
```

2. 承1，請使用物件導向方式實作上題，也就是每個單位的 x 次方做成一個類別後，完成上述儲存功能，並算其結果

Ans:

```
class Term:
    """多項式的一個單項 (如  $6x^4$ )"""
    def __init__(self, coefficient, exponent):
        self.coefficient = coefficient # 係數
        self.exponent = exponent # 次方

    def evaluate(self, x):
        """計算單項的值"""
        return self.coefficient * (x ** self.exponent)

class Polynomial:
    """多項式 (如  $f(x) = 6x^4 + 2x^2 + 3$ )"""
    def __init__(self):
        self.terms = [] # 儲存多項式中的所有單項

    def add_term(self, coefficient, exponent):
        """新增一個單項"""
        self.terms.append(Term(coefficient, exponent))
```

```

def evaluate(self, x):
    """計算多項式的值"""
    return sum(term.evaluate(x) for term in self.terms)

# 建立多項式 f(x) = 6x^4 + 2x^2 + 3
f = Polynomial()
f.add_term(6, 4) # 6x^4
f.add_term(2, 2) # 2x^2
f.add_term(3, 0) # 3 (常數項)

# 計算 x = 91 時的值
x = 91
result = f.evaluate(x)

# 顯示結果
print()
print(f"f({x}) = {result}")

```

3. 在數位圖像處理中，大多數的高分辨率灰度圖像中，很多像素的值為 0，這使得 圖像可以用稀疏矩陣來有效地表示與壓縮。假設你有一個 5x5 的灰度圖像，其中 大多數像素值為 0，只有少數像素有非零值。

```

class SparseMatrix:
    def __init__(self):
        self.data = {}

    def gray(self, i, j, value):
        if value != 0:
            self.data[(i, j)] = value

    def display(self):
        for (i, j), value in self.data.items():
            print(f"Pixel ({i}, {j}) = {value}")

# 建立稀疏矩陣
matrix = SparseMatrix()
matrix.gray(0, 1, 50)
matrix.gray(1, 3, 120)
matrix.gray(2, 4, 180)
matrix.gray(3, 2, 255)

# 顯示結果
matrix.display()

```

4.

Ans:

```
# 讀取輸入
n = int(input("請輸入陣列元素個數：")) # 第一行為陣列元素個數
array = list(map(int, input("請輸入陣列元素：").split())) # 第二行為陣列元素

# 計算逆序對數量
count = 0
for i in range(n):
    for j in range(i + 1, n):
        if array[i] > array[j]:
            count += 1

# 輸出結果
print(f"Output : {count}")
```

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