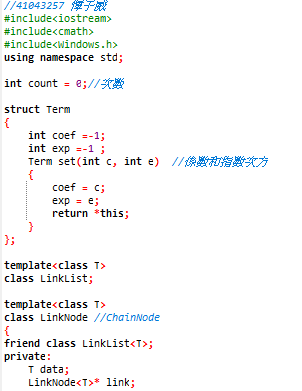
41043257 惲子威

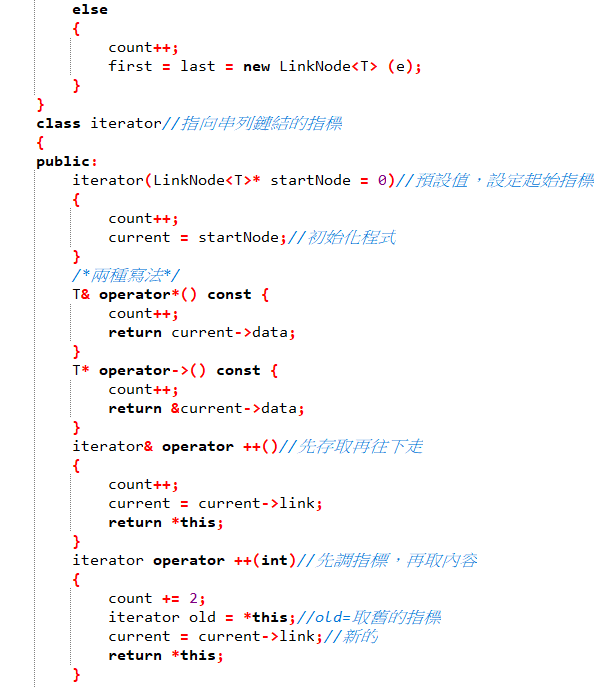
解題說明

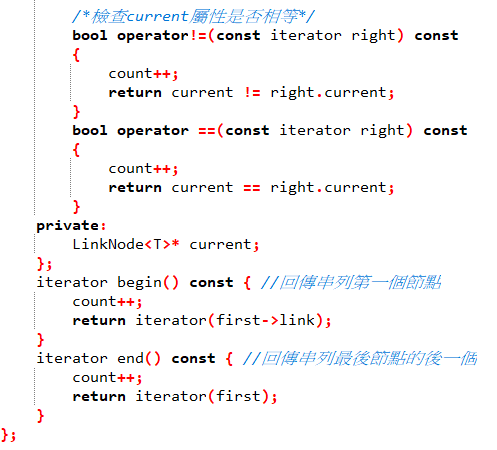
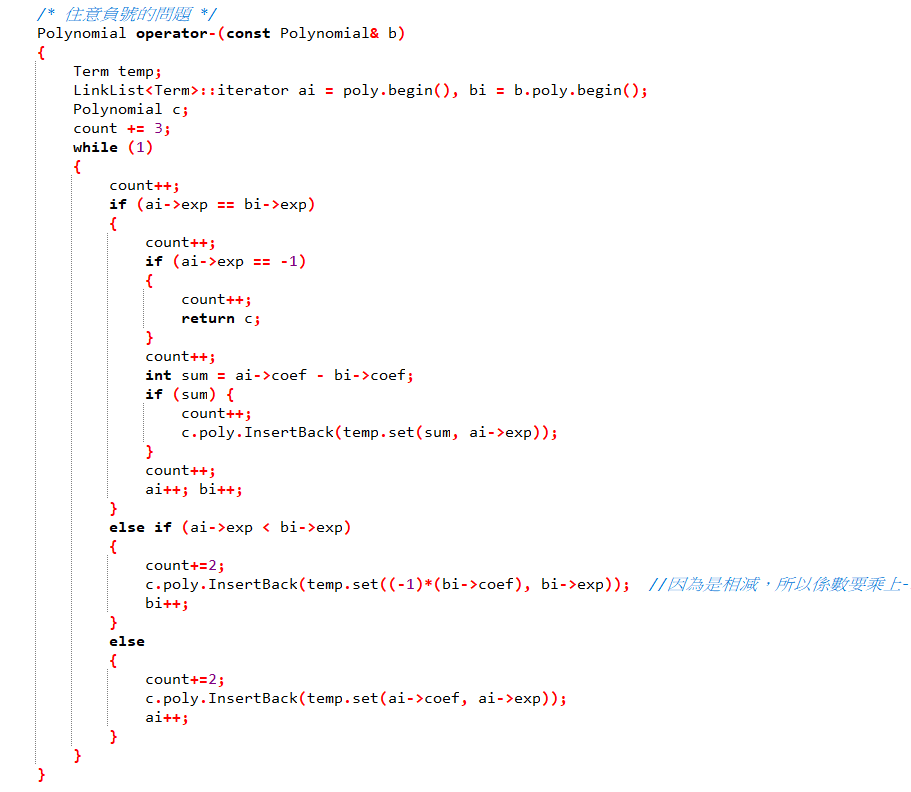
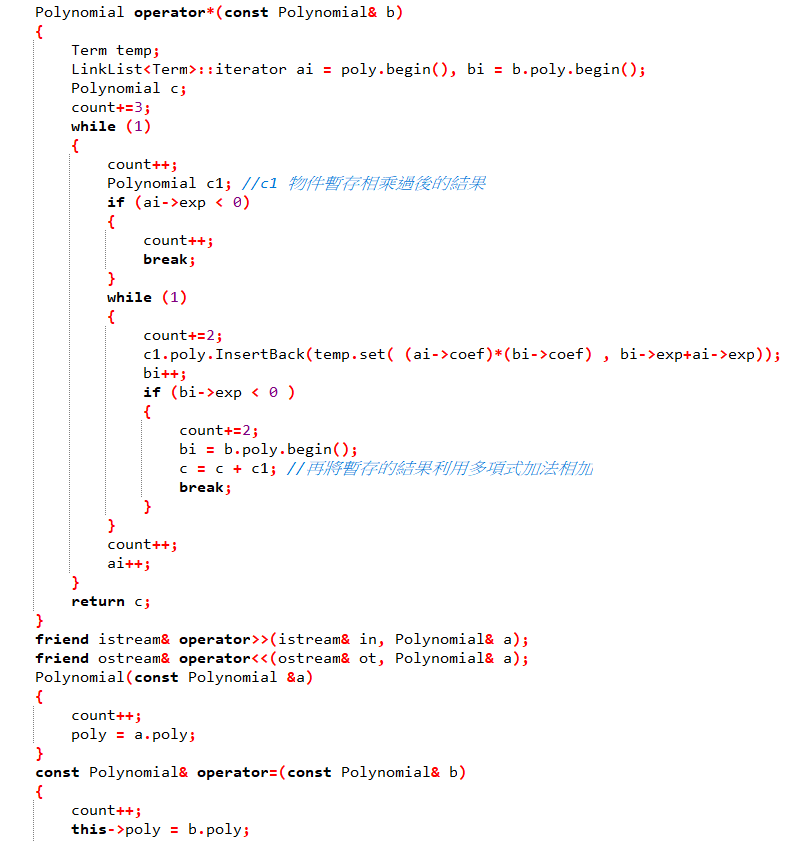
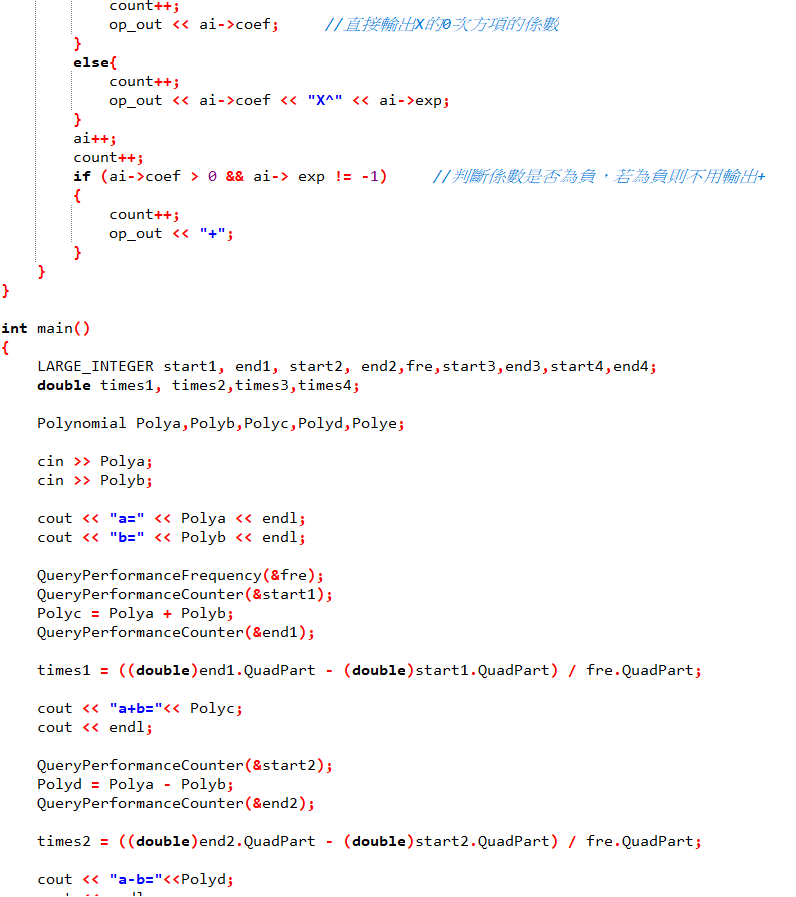
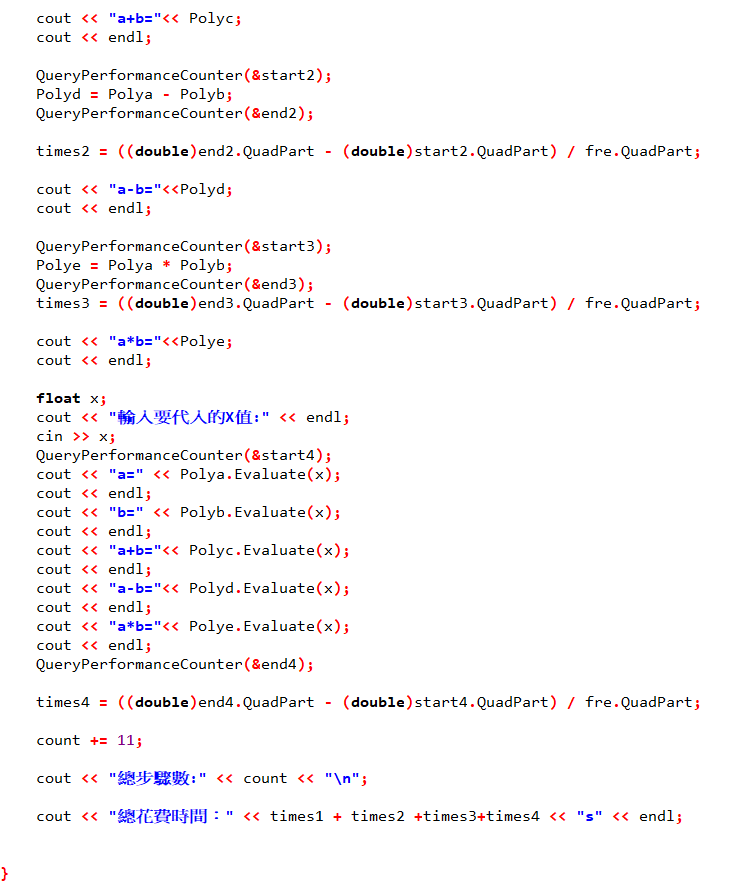
多項式的運算跟上次有一點不同，這次是利用串列鏈結的方式，將多項式的係數與項數表示成一個串列，並利用疊代器控制指標。

Algorithm Design & Programming







Time complexity & Space complex

時間複雜度:

時間複雜度分析取決於Polynomial類別及其相關方法內執行的操作。讓我們分解一下程式碼中關鍵操作的時間複雜度：

**新增 （Polynomial::operator+）**：

加法涉及迭代表示多項式的兩個鍊錶。若兩個多項式分別有m和n項，則加法的時間複雜度為O(m + n)。這是因為兩個多項式中的每一項都被訪問一次。

**減法 ( Polynomial::operator-)** :

與加法類似，減法也涉及兩個鍊錶的迭代。時間複雜度為O(m + n).

**乘法 ( Polynomial::operator\*)** :

兩個多項式的乘法涉及嵌套循環，其中第一個多項式的每一項都乘以第二個多項式的每一項。如果第一個多項式有m項，第二個多項式也有n項，則時間複雜度為O(m \* n)。

**評估 （Polynomial::Evaluate）**：

在特定值處評估多項式x需要迭代多項式的所有項，導致時間複雜度為O(m)，其中m是多項式中的項數。

**空間複雜度**

空間複雜度由儲存多項式所需的記憶體量以及運算所需的任何額外空間決定：

**加法和減法**：

加法或減法的空間複雜度為O(m + n)，其中m和n是兩個多項式的項數。這是因為創建了一個新的多項式來儲存結果。

**乘法**：

乘法的空間複雜度是因為在最壞的情況下O(m \* n)結果多項式可能有最多項。m \* n

**評估**：

評估多項式的空間複雜度是O(1)因為除了幾個變數來儲存總和之外不需要額外的資料結構。

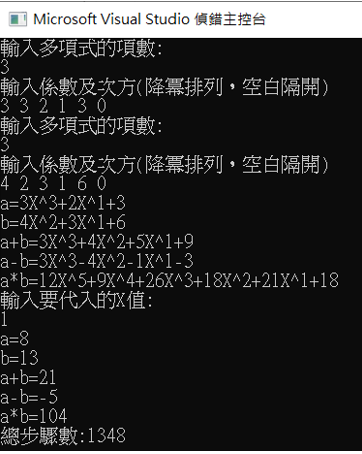
### 整體複雜性總結

**加法與減法**：O(m + n)時間、O(m + n)空間。

**乘法**：O(m \* n)時間、O(m \* n)空間。

**評價**：O(m)時間、O(1)空間。

測試與驗證(Testing and Proving)



輸入：

A=3 3 2 1 3 0🡪3x3+2x1+3

B=4 2 3 1 6 0🡪4x2+3x+6

輸出：

A+B=3x3+4x2+5x+9

A-B=3x3-4x2-x-3

A\*B=(3x3+2x1+3)( 4x2+3x+6)=12x5+9x4+26x3+18x2+21x+18

x帶1：

A=3+2+3=8

B=4+3+6=13

A+B=21

A-B=-5

A\*B=104

心得討論

本次的作業跟上次的題目一樣，但是要用 Circular Linked List 來實作，作法差別很大，但觀察起來，程式的執行效率好了不少。