**四資工二乙 張大軒 41143229 HW3**

**[程式功能]**

1. 資料結構與基本操作:

* + 每個多項式由一個鏈結串列 (Linked List) 表示，串列中的每個節點包含一個係數 (coef)、一個指數 (exp)，以及指向下一個節點的指標 (link)。
  + head 是指向鏈結串列頭節點的指標，這個節點不存放有效數據，只是用來表示鏈結串列的起點。

2. 建構子與解構子:

* Polynomial()：預設建構子，初始化一個空的多項式。
* ~Polynomial()：解構子，釋放鏈結串列中所有節點的記憶體。
* Polynomial(const Polynomial& a)：複製建構子，將另一個多項式的內容複製到新的多項式中。
* const Polynomial& operator=(const Polynomial& a)：賦值運算子，將另一個多項式的內容賦值給當前多項式。

3. 輸入與輸出運算子:

* istream& operator>>(istream& is, Polynomial& x)：重載輸入運算子，允許從輸入流中讀取多項式。
* ostream& operator<<(ostream& os, const Polynomial& x)：重載輸出運算子，將多項式輸出到輸出流中。

4. 多項式的算術運算:

* Polynomial operator+(const Polynomial& b) const：實現兩個多項式的加法。
* Polynomial operator-(const Polynomial& b) const：實現兩個多項式的減法。
* Polynomial operator\*(const Polynomial& b) const：實現兩個多項式的乘法。

5. 多項式的評估:

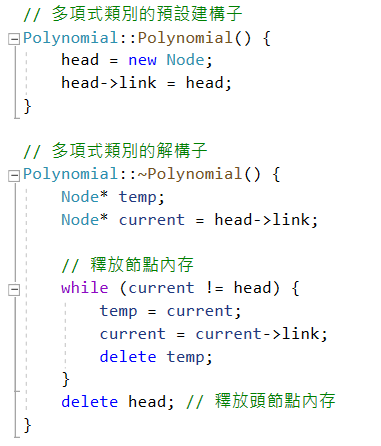
* float Evaluate(float x) const：計算多項式在給定 x 值時的結果。

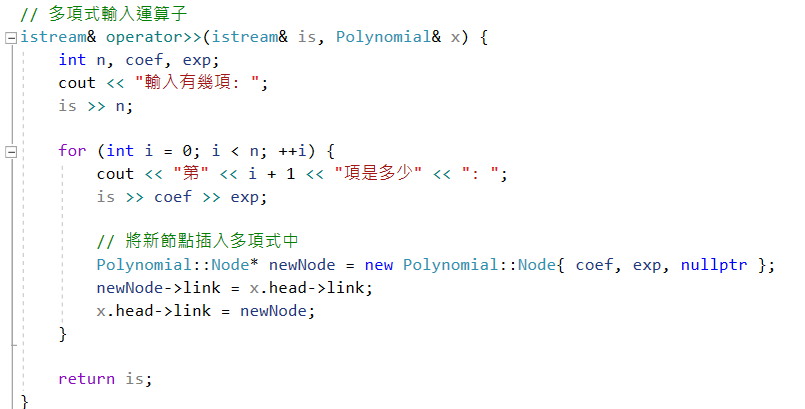
6. 主程式:

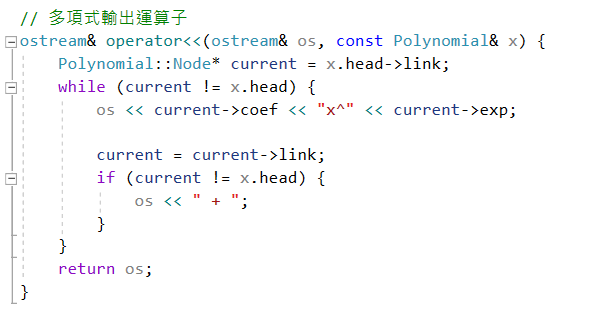
* 主程式用於輸入兩個多項式，並計算它們的和、差、積，然後在給定的 x 值處評估結果。

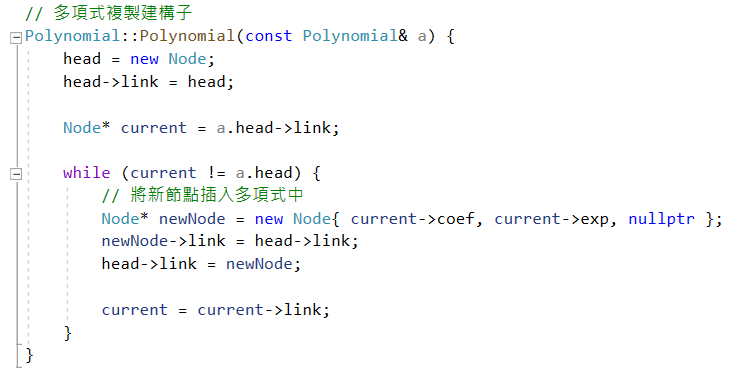
**[程式實作]**





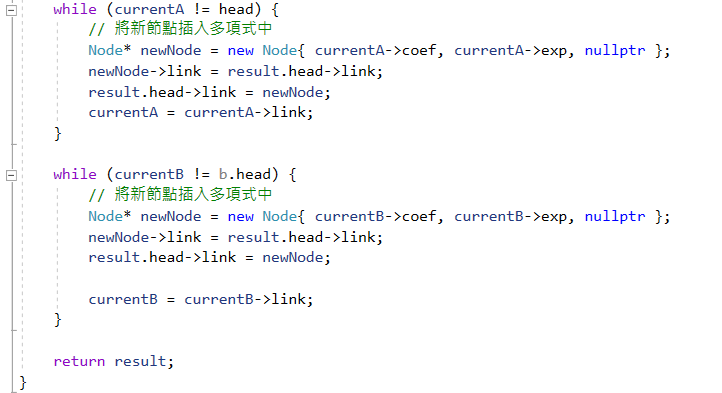




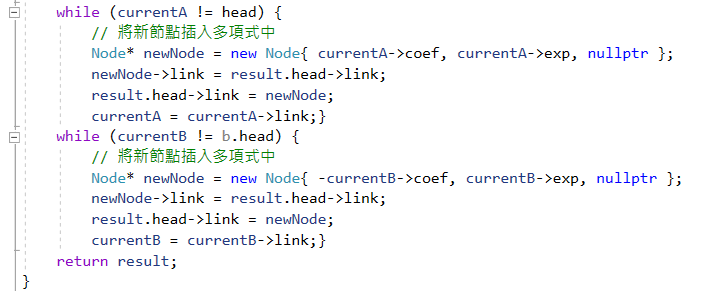




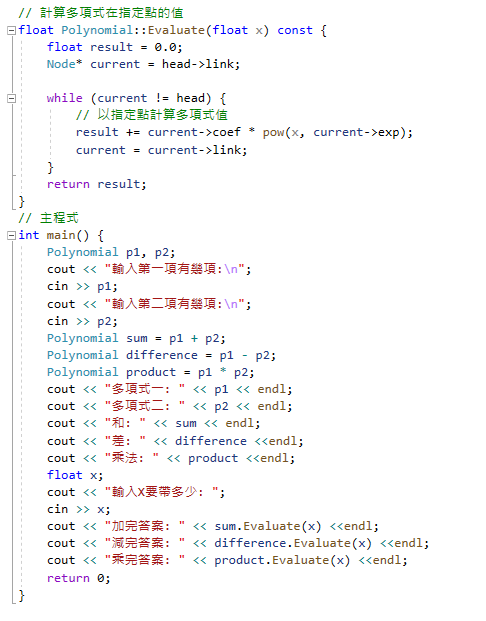




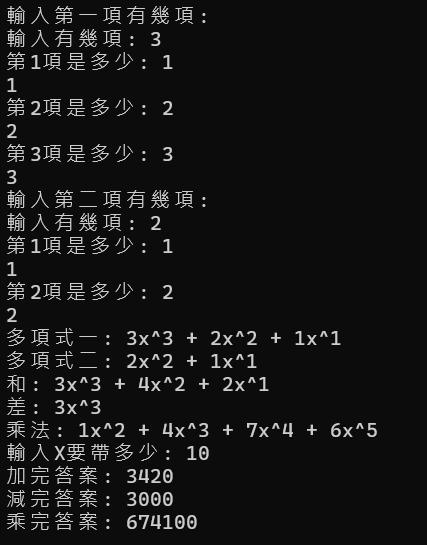








**[程式執行畫面]**



**[時間複雜度]**

* 加、減法 : O(m+n)
* 乘法 : O(m\*n)
* 評估 : O(m)

**[空間複雜度]**

* 加、減法 : O(m+n)
* 乘法 : O(m\*n)
* 評估 : O(1)

**[心得]**

上完暑假這6週的課程，對於資料結構與演算法的概念，比起大二第一次修的時候來的好很多，學到了很多不同的資料排序方式，還有程式實作，雖然程式coding的部分還是相較不熟悉，但經過上機考以及這三次的程式作業，比起之前的自己，有覺得進步很多。