**解題說明**

解題的核心是設計一個多項式類別來進行基本運算（加法、乘法、評估）。首先，定義單項式 (Term) 和多項式 (Polynomial) 的結構，並通過動態陣列存儲多項式的每一項。需要考慮處理多項式運算時指數排序與合併的邏輯，例如進行加法時合併同次項或乘法時正確地計算係數與指數。

**效能分析**

1. 時間複雜度：
   * Add: O(n+m)，其中 n,m為兩多項式的項數。
   * Mult: O(n⋅m)。
   * Eval: O(n)。
2. 空間複雜度：動態陣列需求為O(n)。

**測試與驗證**

測試範例：

1. 測試加法：  
   輸入 p(x)=2x^2+3x，q(x) = x^3+x+1。  
   預期輸出 p(x)+q(x)=x^3+2x^2+4x+1。
2. 測試乘法：  
   輸入 p(x)=2x^2+1，q(x)=x+1。  
   預期輸出 p(x)⋅q(x)=2x^3+2x^2+x+1。
3. 測試評估：  
   p(x)=x^2+x+1，代入 x=2預期輸出7。

**效能量測**

測試資料規模增加（如項數 n=1000,10000）時記錄時間與記憶體消耗，驗證演算法隨規模增長的效能表現。

**心得討論**

這個程式主要設計了一個處理多項式運算的架構，透過物件導向的方式建立 Term 和 Polynomial 類別，清楚地將單項式和多項式的邏輯分開處理。過程中最大的挑戰在於確保動態記憶體管理的正確性，以及處理多項式加法和乘法時，指數相同項目的合併。程式設計還需要考量未來擴展的彈性，例如動態陣列的擴容及保護私有成員的存取。完成這個專案後，我更深刻地理解了C++中類別設計、記憶體管理及運算子多載的重要性，也體會到細節對於程式正確性和效能的關鍵影響。