

```
Microsoft Visual Studio 偵錯1 × + -
請依序輸入多項式的係數與次方數 (結束輸入以換行) : 1 3 2 1 3 0
第一個多項式為 :  $1x^3 + 2x^1 + 3x^0$ 
請依序輸入第二個多項式的係數與次方數 (結束輸入以換行) : 3 2 1 0
第二個多項式為 :  $3x^2 + 1x^0$ 
兩多項式相加結果為 :  $1x^3 + 3x^2 + 2x^1 + 4x^0$ 
兩多項式相乘結果為 :  $3x^5 + 7x^3 + 2x^1 + 9x^2 + 3x^0$ 
請輸入一個x值代入第一個多項式進行計算 : 2
計算結果為 : 15

C:\Users\2014o\OneDrive\桌面\VS\dataHW2\Project1\Debug\Project1.exe (處理序 20092) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [偵錯停止時，自動關閉主控台]。
按任意鍵關閉此視窗...
```

# Polynomial.cpp

## Polynomial():

- 建構子，構造函數會初始化 `capacity` 為 10, `terms` 為 0, 並為 `termArray` 分配內存。

## Resize():

- 這個函數會在當前的 `termArray` 容量不足時，將容量加倍，並將舊的項目移到新的陣列中。
- 這樣的處理使得我們可以動態擴充 `termArray`, 防止溢出。

## Add():

- 這個函數的目的是將兩個多項式相加。
- 我們逐個比對兩個多項式的項目：
  - 如果次方數相同，則將係數相加。
  - 如果次方數不同，則直接將項目加入結果多項式。
- 若相加的結果為零，則跳過該項目。

## Mult():

- 這個函數實現了多項式的乘法。
- 我們遍歷兩個多項式的所有項目，對每一對項目進行相乘，並將結果項目加入 `result`。
- 如果結果的次方數已經存在於 `result` 中，就將係數累加，否則就新增項目。

## Eval():

- 這個函數用來計算多項式在某個特定  $x$  值下的結果。
- 我們使用了 `pow(f, exp)` 計算每個項目的  $x$  次方，並將每項的結果累加起來。

# Operator.cpp

## `operator<<:`

- 這個函數實現了輸出多項式的功能。
- 我們遍歷 `Polynomial` 中的所有項目，將它們按照係數 $x^{\text{次方數}}$  的格式輸出。
- 如果有多個項目，我們會在每個項目後加上 `+`，除非是最後一個項目。

## `operator>>:`

- 這個函數實現了從標準輸入讀取多項式的功能。
- 我們使用一個 `while` 迴圈來反覆讀取用戶輸入的係數和次方數。
- 若讀取的係數或次方數無效(例如空格或換行)，則退出循環。
- 在每次讀取新項目之前，我們檢查 `termArray` 是否已滿，若滿了則調用 `Resize()` 增加容量。