

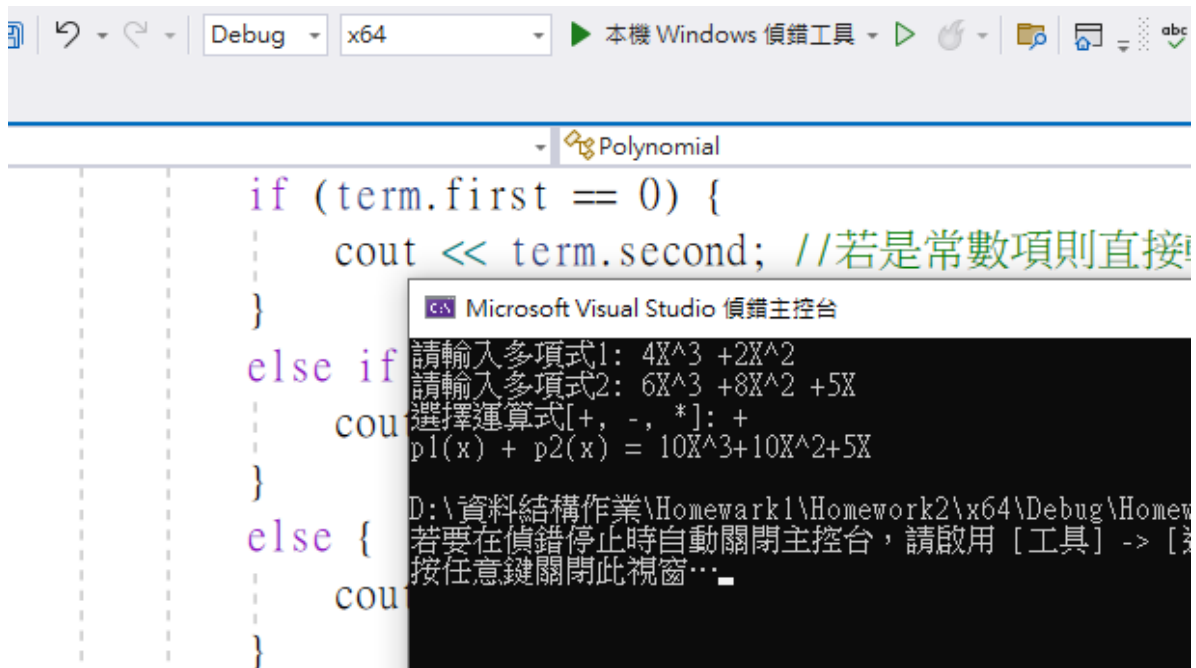
# Homework2

多項式的加減成

解題思路:

多項式是由很多的係數與指數所組成，所以先把要運算的兩組多項式的指數與係數分開來儲存，等判斷好輸入者要的運算符號時就可以把兩邊指數一樣的係數相 $+-*$ ，然後再把答案依照大小以及正負輸出出來。

執行結果測試:



The screenshot shows a Visual Studio IDE in Debug mode for an x64 architecture. The active file is named 'Polynomial'. The code in the editor is as follows:

```
if (term.first == 0) {  
    cout << term.second; //若是常數項則直接  
}  
else if  
    cou  
}  
else {  
    cou  
}
```

The console window, titled 'Microsoft Visual Studio 偵錯主控台', displays the following text:

```
請輸入多項式1: 4X^3 +2X^2  
請輸入多項式2: 6X^3 +8X^2 +5X  
選擇運算式[+, -, *]: +  
p1(x) + p2(x) = 10X^3+10X^2+5X  
  
D:\資料結構作業\Homework1\Homework2\x64\Debug\Homew  
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [  
按任意鍵關閉此視窗...
```

效能分析:

時間複雜度:

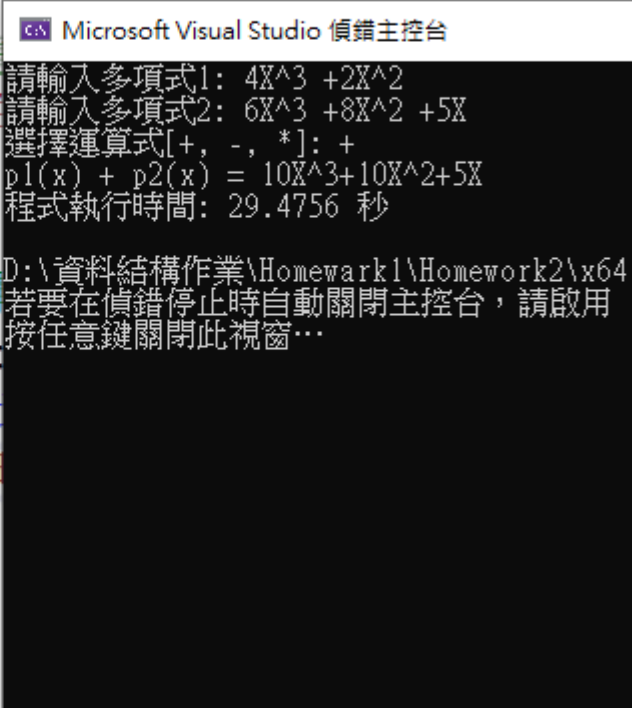
- addterm 的時間複雜度是  $O(\log n)$ 。
- print 的時間複雜度是  $O(n)$ 。
- add、sub 函數的時間複雜度是  $O(n+m)$ 。
- mult 函數的時間複雜度是  $O(n*m)$ 。
- check 函數的時間複雜度是  $O(L*\log n)$ 。

空間複雜度:

主要由 map 結構中的項目數量決定。

效能實測:

```
result = p1.mult(p2);
cout << "p1(x) * p2(x) = ";
break;
default://無效的運算
cout << "無效的運算";
return 1;
}
result.print();//輸出結果
auto end = chrono::now();
chrono::duration<chrono::high_resolution_clock::time_point> duration = end - start;
cout << "程式執行時間: " << duration.count() << "秒";
return 0;
}
```



The screenshot shows the Microsoft Visual Studio console window. The title bar reads "Microsoft Visual Studio 偵錯主控台". The console output is as follows:  
請輸入多項式1: 4X^3 +2X^2  
請輸入多項式2: 6X^3 +8X^2 +5X  
選擇運算式[+, -, \*]: +  
p1(x) + p2(x) = 10X^3+10X^2+5X  
程式執行時間: 29.4756 秒  
Below the output, there is a path: D:\資料結構作業\Homework1\Homework2\x64  
And a message: 若要在偵錯停止時自動關閉主控台，請啟用  
按任意鍵關閉此視窗...

心得:

一開始寫程享用陣列直接完成，結果後來發現直接用陣列寫的話可能會因為多項式太長就爆開所以改用其他方式記錄，雖然題目不難但有時候會因為紀錄太多東西搞亂了到

底哪個要與哪相+-\*還是指數紀錄指標要移動，有點麻煩。