

因為我不會讀取好幾個字串，所以我讓使用者輸入要輸入幾個多項式的成員，並丟進一個名為“input”的函式去讀取多項式的係數和次方，讀取完後丟入“termArray”的陣列，並把係數和次方儲存起來。並且可讓使用者設定 x 的值，然後輸出如圖

```
Input first polynomial:
Enter number of terms: 1
Enter coefficient and exponent for term 1: 1 0
Input second polynomial:
Enter number of terms: 3
Enter coefficient and exponent for term 1: 10 0
Enter coefficient and exponent for term 2: 5 6
Enter coefficient and exponent for term 3: 1 1
First polynomial:  $1x^0$ 
Second polynomial:  $10x^0 + 5x^6 + 1x^1$ 
Sum:  $11x^0 + 5x^6 + 1x^1$ 
Product:  $10x^0 + 5x^6 + 1x^1$ 
Enter a value to evaluate the polynomials: 2
First polynomial evaluated at 2: 1
Second polynomial evaluated at 2: 332
Sum evaluated at 2: 333
Product evaluated at 2: 332
```

效能量測

```
Input first polynomial:
Enter number of terms: 1
Enter coefficient and exponent for term 1: 1 0
Input second polynomial:
Enter number of terms: 3
Enter coefficient and exponent for term 1: 10 0
Enter coefficient and exponent for term 2: 5
6
Enter coefficient and exponent for term 3: 1 1
Time taken for addition: 2 microseconds
Time taken for multiplication: 1 microseconds
First polynomial:  $1x^0$ 
Second polynomial:  $10x^0 + 5x^6 + 1x^1$ 
Sum:  $11x^0 + 5x^6 + 1x^1$ 
Product:  $10x^0 + 5x^6 + 1x^1$ 
Enter a value to evaluate the polynomials: 2
First polynomial evaluated at 2: 1
Second polynomial evaluated at 2: 332
Sum evaluated at 2: 333
Product evaluated at 2: 332
```

效能分析

1. Polynomial::Add(Polynomial poly):

- 合併兩個多項式的過程中，每個多項式的項數分別為 `terms` 和 `poly.terms`。
- 合併相同指數的項的過程中，最壞情況下需要比較 `terms + poly.terms` 次。
- 因此，時間複雜度為 $O(n + m)$ ，其中 n 是第一個多項式的項數， m 是第二個多項式的項數。

2. Polynomial::Mult(Polynomial poly):

- 兩個多項式相乘的過程中，每個多項式的項數分別為 `terms` 和 `poly.terms`。
- 每個項與另一個多項式的每個項相乘，總共需要 `terms * poly.terms` 次。
- 合併相同指數的項的過程中，最壞情況下需要比較 `terms * poly.terms` 次。
- 因此，時間複雜度為 $O(n * m)$ ，其中 n 是第一個多項式的項數， m 是第二個多項式的項數。

3. Polynomial::Mult(Polynomial poly):

- 兩個多項式相乘的過程中，每個多項式的項數分別為 `terms` 和 `poly.terms`。
- 每個項與另一個多項式的每個項相乘，總共需要 `terms * poly.terms` 次。
- 合併相同指數的項的過程中，最壞情況下需要比較 `terms * poly.terms` 次。
- 因此，時間複雜度為 $O(n * m)$ ，其中 n 是第一個多項式的項數， m 是第二個多項式的項數。

4. Polynomial::Input():

- 輸入多項式的過程中，需要輸入每一項的係數和指數。
- 因此，時間複雜度為 $O(n)$ ，其中 n 是多項式的項數。

5. Polynomial::Print() const:

- 輸出多項式的過程中，需要遍歷多項式的每一項並輸出。
- 因此，時間複雜度為 $O(n)$ ，其中 n 是多項式的項數。

Add 方法的時間複雜度為 $O(n + m)$ 。

Mult 方法的時間複雜度為 $O(n * m)$ 。

Eval 方法的時間複雜度為 $O(n)$ 。

Input 方法的時間複雜度為 $O(n)$ 。

Print 方法的時間複雜度為 $O(n)$ 。

心得討論

因為還不會讀取複數字串，導致程式靈活度不高，若學會可增加程式靈活度!