

想法:

Add 先判斷指數是否一樣,一樣相加

Subtraction 先判斷指數是否一樣,一樣相減

Mult 一項一項抓出來,係數相乘,次方相加,遇到一樣次方的就把係數加起來

放進去,不然就開一格放

效能分析

```

Polynomial Polynomial::Mult(Polynomial b){
    Polynomial c;
    int aPos = 0;
    c.termArray = new Term[capacity];
    write ((aPos < terms));
    for (int bPos = 0 ; bPos < b.terms ; bPos++){
        int ok = 0;
        float t = termArray[aPos].coef * b.termArray[bPos].coef;
        int y = termArray[aPos].exp + b.termArray[bPos].exp;
        for (int i = 0; i < c.terms; i++) {
            if (c.termArray[i].exp == y) {
                c.termArray[i].exp = y;
                c.termArray[i].coef += t;
                ok = 1;
            }
        }
        if (ok == 0) {
            c.NewTerm(t, y);
        }
    }
    aPos++;
}

for (int i = 0; i < c.terms; i++) {
    for (int j = (i + 1) ; j < c.terms ; j++){
        if (c.termArray[i].exp < c.termArray[j].exp){
            int tempExp = c.termArray[i].exp;
            c.termArray[i].exp = c.termArray[j].exp;
            c.termArray[j].exp = tempExp;

            float tempCoef = c.termArray[i].coef;
            c.termArray[i].coef = c.termArray[j].coef;
            c.termArray[j].coef = tempCoef;
        }
    }
}

return c;
}

```

$O(1)$

$O(N^2)$

//downgrade

$O(N^2)$

測試與驗證

```
C:\Users\USER\Downloads\41 X + | v
請輸入第一個多項式：
請輸入多項式的項數：4
請輸入係數與指數：4 2
請輸入係數與指數：3 6
請輸入係數與指數：2 5
請輸入係數與指數：8 4
請輸入第二個多項式：
請輸入多項式的項數：3
請輸入係數與指數：2 5
請輸入係數與指數：6 3
請輸入係數與指數：7 2
第一個多項式：4x^2 + 3x^6 + 2x^5 + 8x^4
第二個多項式：2x^5 + 6x^3 + 7x^2
兩個多項式的和：2x^5 + 6x^3 + 11x^2 + 3x^6 + 2x^5 + 8x^4
兩個多項式的積：70x^7 + 24x^5 + 28x^4 + 6x^11 + 34x^9 + 33x^8 + 4x^10 + 56x^6
請輸入 x 的值以計算第一個多項式的值：|
```

開發報告

開發過程

需求分析：

1. 設計一個多項式類別，實現加法、減法與乘法運算，並能以直觀方式輸入與輸出。

系統設計：

1. 採用分層設計，將多項式的運算邏輯與單項式的資料儲存分離。
2. 多項式的邏輯主要集中在 Polynomial 類別內，Term 類別僅作為資料存儲單元。

心得討論

完成題目後我對 class 的使用,跟運算子多載又更了解了

申論

該程式已能穩定地執行多項式的基本運算，具備良好的擴展性與結構清晰的設計。仍有改進空間，例如提升效能與擴充功能範圍。透過未來的持續優化，可進一步提升此程式的實用性與表現。