資料結構報告

林聖紘

Jan. 07 / 2025

Polynomial

1 解題說明 2
2 演算法設計與設計 3
3 效能分析 8
4 測試與過程 9

解題說明

使用**鏈結串列**（Linked List）來表示多項式，每個節點包含：

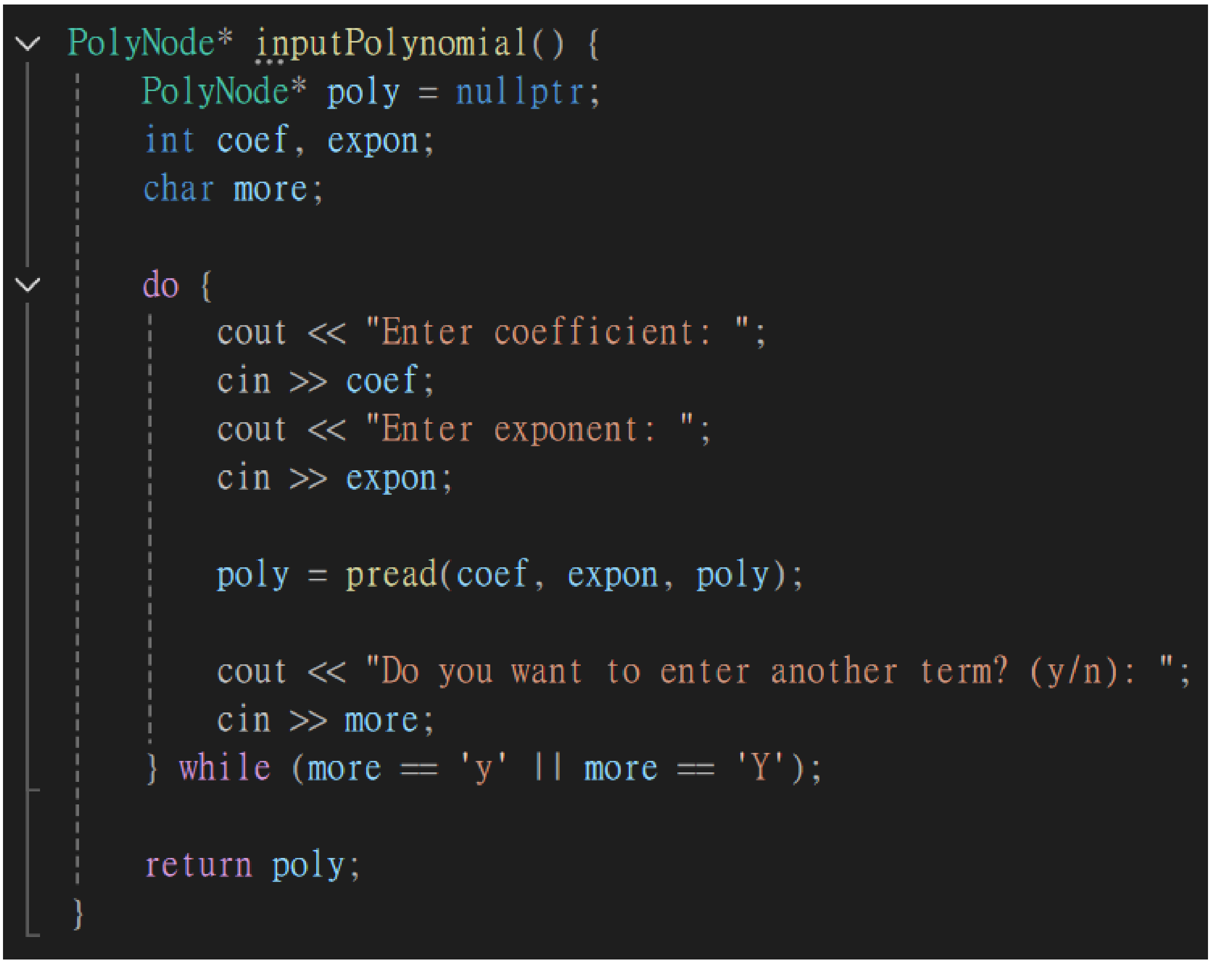
- coef：係數。
- expon：指數。
- link：指向下一個節點的指標。

演算法設計與設計

輸入與輸出

輸入：

使用者輸入多項式的每一項，指定其係數和指數，並以鏈結串列形式儲存。

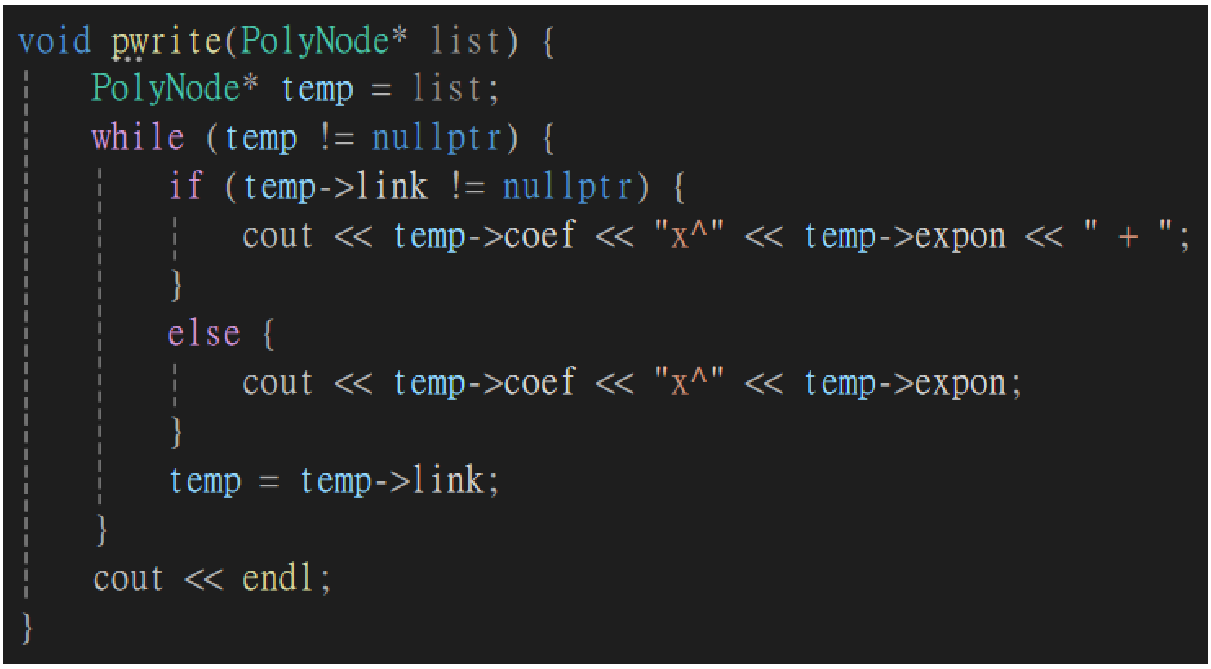


```cpp
PolyNode* inputPolynomial() {
    PolyNode* poly = nullptr;
    int coef, expon;
    char more;

    do {
        cout << "Enter coefficient: ";
        cin >> coef;
        cout << "Enter exponent: ";
        cin >> expon;

        poly = pread(coef, expon, poly);

        cout << "Do you want to enter another term? (y/n): ";
        cin >> more;
    } while (more == 'y' || more == 'Y');

    return poly;
}
```

演算法設計與設計

輸出：

多項式的加法、減法和乘法結果。



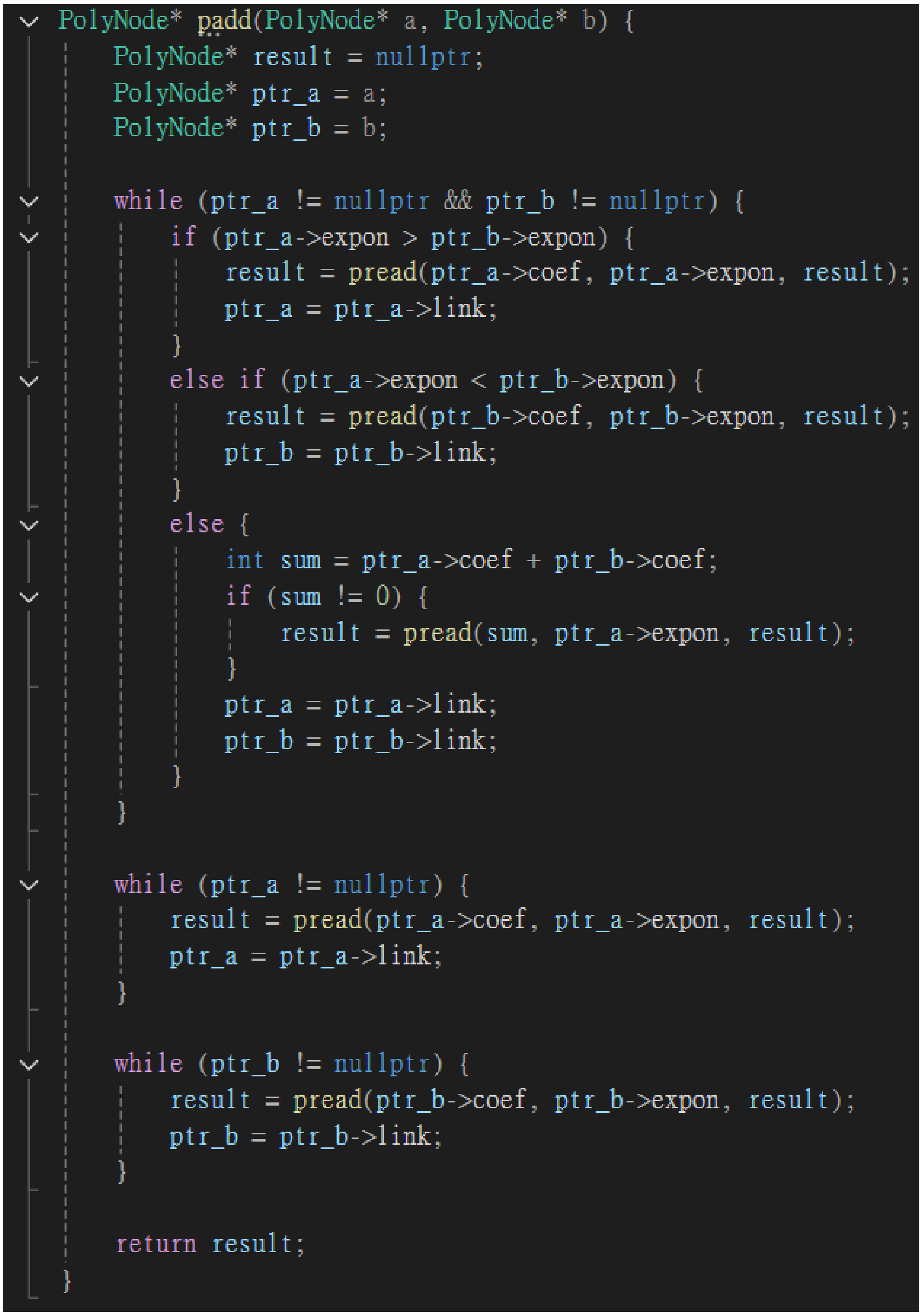
```cpp
void pwrite(PolyNode* list) {
    PolyNode* temp = list;
    while (temp != nullptr) {
        if (temp->link != nullptr) {
            cout << temp->coef << "x^" << temp->expon << " + ";
        }
        else {
            cout << temp->coef << "x^" << temp->expon;
        }
        temp = temp->link;
    }
    cout << endl;
}
```

演算法設計與設計

運算實現

執行多項式加法、減法或乘法，生成結果。

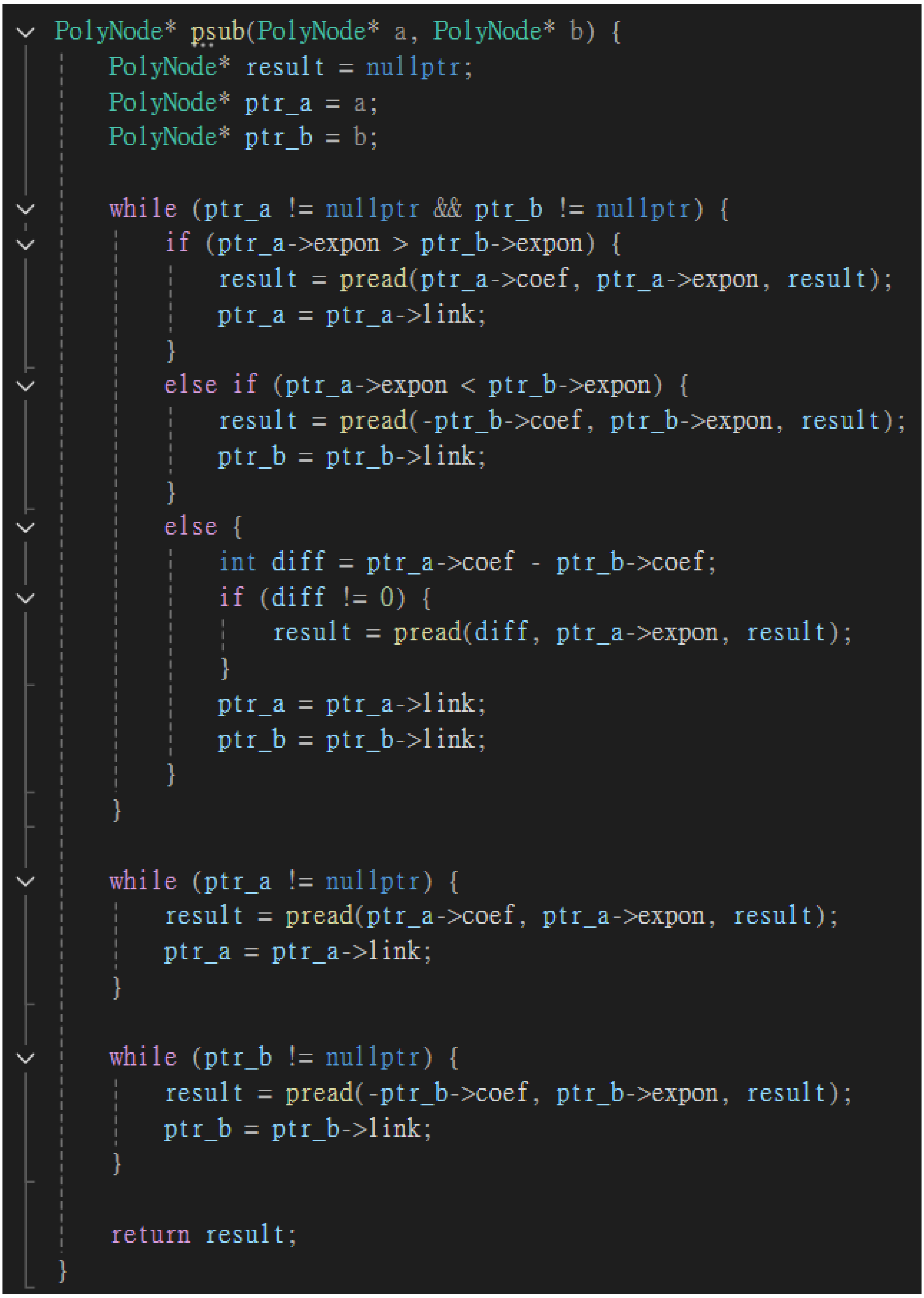
加法:



```
PolyNode* padd(PolyNode* a, PolyNode* b) {
    PolyNode* result = nullptr;
    PolyNode* ptr_a = a;
    PolyNode* ptr_b = b;

    while (ptr_a != nullptr && ptr_b != nullptr) {
        if (ptr_a->expon > ptr_b->expon) {
            result = pread(ptr_a->coef, ptr_a->expon, result);
            ptr_a = ptr_a->link;
        }
        else if (ptr_a->expon < ptr_b->expon) {
            result = pread(ptr_b->coef, ptr_b->expon, result);
            ptr_b = ptr_b->link;
        }
        else {
            int sum = ptr_a->coef + ptr_b->coef;
            if (sum != 0) {
                result = pread(sum, ptr_a->expon, result);
            }
            ptr_a = ptr_a->link;
            ptr_b = ptr_b->link;
        }
    }

    while (ptr_a != nullptr) {
        result = pread(ptr_a->coef, ptr_a->expon, result);
        ptr_a = ptr_a->link;
    }

    while (ptr_b != nullptr) {
        result = pread(ptr_b->coef, ptr_b->expon, result);
        ptr_b = ptr_b->link;
    }

    return result;
}
```

演算法設計與設計

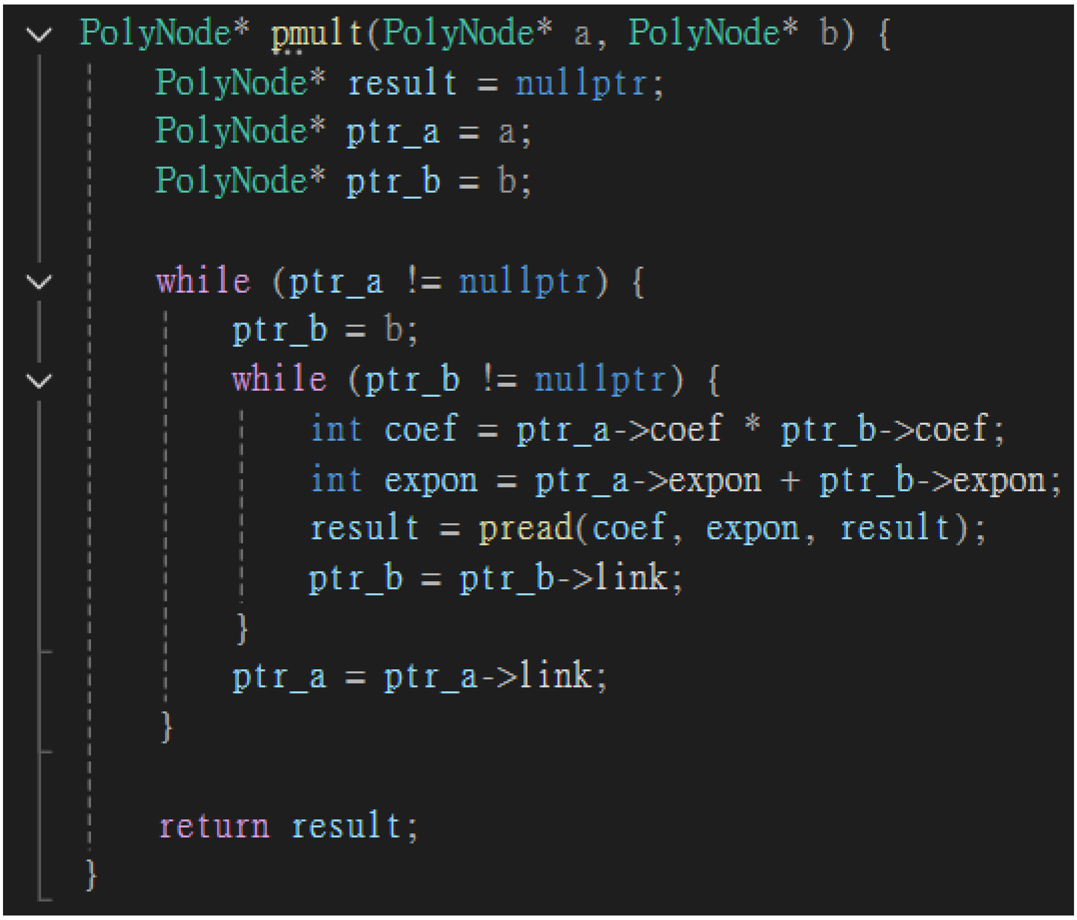
減法:



```
PolyNode* psub(PolyNode* a, PolyNode* b) {
    PolyNode* result = nullptr;
    PolyNode* ptr_a = a;
    PolyNode* ptr_b = b;

    while (ptr_a != nullptr && ptr_b != nullptr) {
        if (ptr_a->expon > ptr_b->expon) {
            result = pread(ptr_a->coef, ptr_a->expon, result);
            ptr_a = ptr_a->link;
        }
        else if (ptr_a->expon < ptr_b->expon) {
            result = pread(-ptr_b->coef, ptr_b->expon, result);
            ptr_b = ptr_b->link;
        }
        else {
            int diff = ptr_a->coef - ptr_b->coef;
            if (diff != 0) {
                result = pread(diff, ptr_a->expon, result);
            }
            ptr_a = ptr_a->link;
            ptr_b = ptr_b->link;
        }
    }

    while (ptr_a != nullptr) {
        result = pread(ptr_a->coef, ptr_a->expon, result);
        ptr_a = ptr_a->link;
    }

    while (ptr_b != nullptr) {
        result = pread(-ptr_b->coef, ptr_b->expon, result);
        ptr_b = ptr_b->link;
    }

    return result;
}
```

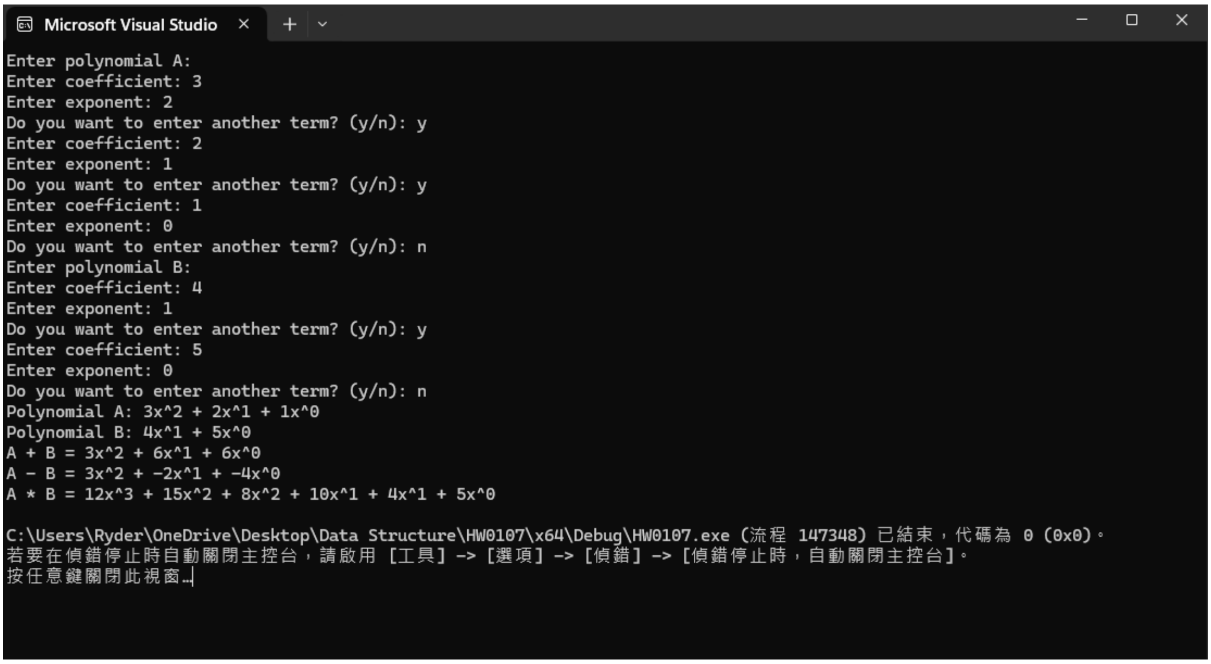
演算法設計與設計

乘法:



```
PolyNode* pmult(PolyNode* a, PolyNode* b) {
    PolyNode* result = nullptr;
    PolyNode* ptr_a = a;
    PolyNode* ptr_b = b;

    while (ptr_a != nullptr) {
        ptr_b = b;
        while (ptr_b != nullptr) {
            int coef = ptr_a->coef * ptr_b->coef;
            int expon = ptr_a->expon + ptr_b->expon;
            result = pread(coef, expon, result);
            ptr_b = ptr_b->link;
        }
        ptr_a = ptr_a->link;
    }

    return result;
}
```

效能分析



```
Enter polynomial A:
Enter coefficient: 3
Enter exponent: 2
Do you want to enter another term? (y/n): y
Enter coefficient: 2
Enter exponent: 1
Do you want to enter another term? (y/n): y
Enter coefficient: 1
Enter exponent: 0
Do you want to enter another term? (y/n): n
Enter polynomial B:
Enter coefficient: 4
Enter exponent: 1
Do you want to enter another term? (y/n): y
Enter coefficient: 5
Enter exponent: 0
Do you want to enter another term? (y/n): n
Polynomial A: 3x^2 + 2x^1 + 1x^0
Polynomial B: 4x^1 + 5x^0
A + B = 3x^2 + 6x^1 + 6x^0
A - B = 3x^2 + -2x^1 + -4x^0
A * B = 12x^3 + 15x^2 + 8x^2 + 10x^1 + 4x^1 + 5x^0

C:\Users\Ryder\OneDrive\Desktop\Data Structure\HW0107\x64\Debug\HW0107.exe (流程 147348) 已結束，代碼為 0 (0x0)。
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…
```

測試與過程

式實現了多項式的加法、減法和乘法運算，使用鏈結串列（Linked List）來儲存多項式的每一項。藉由讓使用者輸入多項式的係數和指數，程式可以動態地建立多項式，並進行運算後輸出結果。