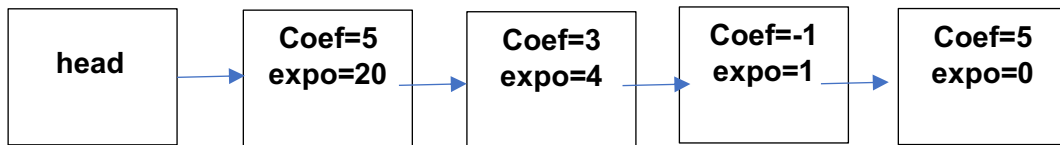


การจำลองการทำงานของ Polynomial

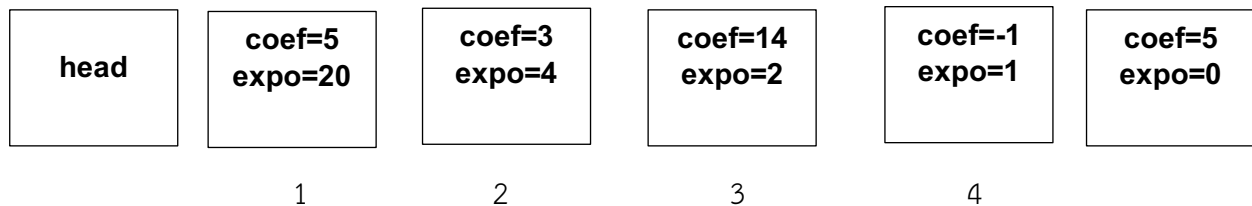
กำหนดโครงสร้างของ Class Polynomial ซึ่งทำหน้าที่เสมือนการเก็บข้อมูล Polynomial (สามารถ Download ได้จาก Github ตามขั้นตอนการทำการบ้านด้านล่าง) แต่ละ Polynomial ก็จะมีแต่ละเทอม เช่น $f1 = 5X^{20} + 3X^4 - X + 5$ Class นี้จะเก็บข้อมูลลักษณะของพจน์ต่อกันตามลำดับจากพจน์ที่กำลังมากไปน้อยดังรูป



อย่างไรก็ตามเพื่อความสะดวกเวลาแสดงผลจะแสดงในลักษณะนี้ $[5X^{20} \ 3X^4 \ -1X^1 \ 5X^0]$ ทั้งนี้พจน์ที่ coef เป็น 0 จะไม่แสดง(implement ให้แล้ว)

หน้าที่ของเราคือเขียน Method ต่อไปนี้โดย implement แบบ Pointer-based

- **void addTerm (int coef, int exponent)** เป็น method ที่เพิ่มพจน์ใหม่ให้กับ Polynomial โดยที่เมื่อเพิ่มพจน์แล้วจะต้องเรียงจากมากไปน้อยตามเลขยกกำลัง เช่น เมื่อเรียก addTerm(14,2) ต่อจาก Polynomial ข้างต้น จะได้ว่า มีพจน์ $14X^2$ แทรกระหว่าง $3X^4$ และ $-X$



หมายเหตุ อาจจะมีพจน์ที่เป็นบวก พจน์ที่เป็นลบ พจน์ที่เลขยกกำลังเท่ากับที่เคยมีใน Polynomial แล้วก็ได้

- **void plus(Polynomial f2)** เป็น method ที่บวก Polynomial f2 เข้ากับ Polynomial ที่เรียก ตัวอย่างเช่น สมมติว่า

$$f1 = 5X^8 + 4X^3 + 2X \text{ (แสดงผลเป็น [} 5X^8 \ 4X^3 \ 2X^1 \text{])}$$

$$f2 = 4X^5 + X - 5 \text{ (แสดงผลเป็น [} 4X^5 \ 1X^1 \ -5X^0 \text{])}$$

$$\text{หลังเรียก } f1.plus(f2) \text{ จะได้ } f1 = 5X^8 + 4X^5 + 4X^3 + 3X - 5$$

$$\text{(แสดงผลเป็น [} 5X^8 \ 4X^5 \ 4X^3 \ 3X^1 \ -5X^0 \text{])}$$

`void minus(Polynomial f2)` เป็น method ที่ลบ Polynomial f2 ออกจาก Polynomial ที่เรียก ตัวอย่างเช่น สมมติว่า

$$f1 = 5X^8 + 4X^3 + 2X \text{ (แสดงผลเป็น [} 5X^{\{8\}} \ 4X^{\{3\}} \ 2X^{\{1\}} \text{])}$$

$$f2 = 4X^5 + X - 5 \text{ (แสดงผลเป็น [} 4X^{\{5\}} \ 1X^{\{1\}} \ -5X^{\{0\}} \text{])}$$

- หลังเรียก `f1.minus(f2)` จะได้ $f1 = 5X^8 - 4X^5 + 4X^3 + X + 5$
(แสดงผลเป็น [$5X^{\{8\}} \ -4X^{\{5\}} \ 4X^{\{3\}} \ 1X^{\{1\}} \ 5X^{\{0\}}$])

ขั้นตอนการทำงานบ้าน

1. ดาวน์โฮลด์ไฟล์การบ้านได้ที่ <https://github.com/CS-CMU/cs252student>
2. ในโฟลเดอร์ HW03 จะมีไฟล์ `main.cpp`, `term.cpp` และ `HW03.cpp`
3. ไฟล์ `main.cpp` จะเป็นไฟล์สำหรับทดสอบโปรแกรมของนักศึกษา สามารถแก้ไขได้ตามความเหมาะสม ไฟล์นี้ไม่ต้องส่งและไม่มีการตรวจ
4. ไฟล์ `term.cpp` มีนิยามของคลาส `Poly_node` โดยมี
 - ตัวแปร `int coef` ที่เก็บสัมประสิทธิ์
 - ตัวแปร `int exponent` ที่เก็บเลขชี้กำลัง
 - ตัวแปร `Poly_node * next` ที่ชี้ไปยังพจน์ถัดไป ไฟล์นี้ห้ามแก้ไข แม้จะไม่ต้องส่งก็ตาม
5. ไฟล์ `HW03.cpp` จะเป็นไฟล์ที่นักศึกษาต้องแก้ไข โดยมีรายละเอียดดังนี้

คลาส `Polynomial` ประกอบไปด้วย

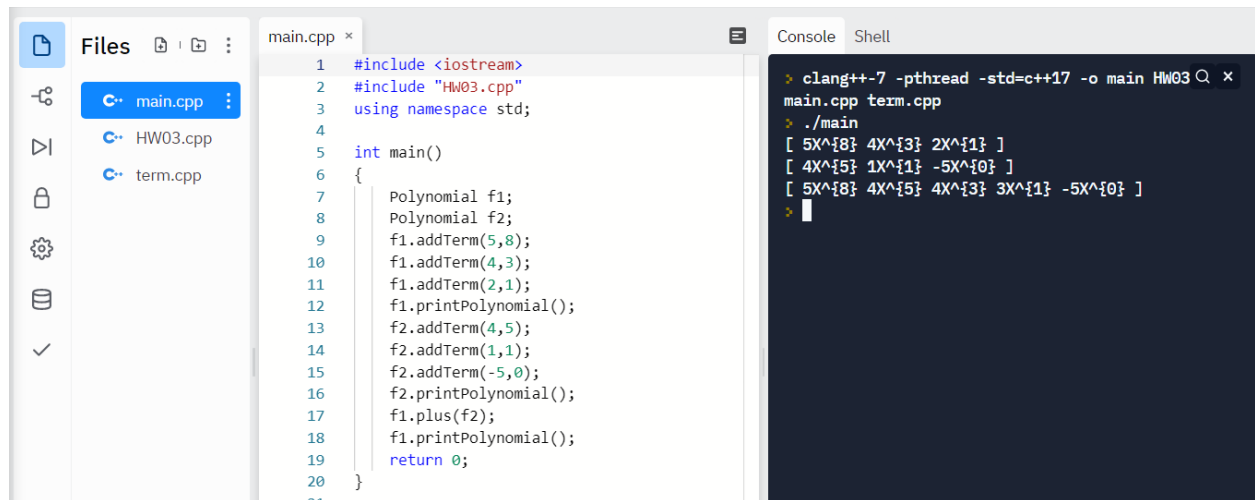
1. ตัวแปร Pointer `Poly_node * head` ตัวแปรตัวนี้ต้องประกาศเป็น `public` เท่านั้น (ใช้สำหรับการตรวจ)
2. Constructor สำหรับคลาส `Polynomial` ให้ค่าเริ่มต้นกับ Pointer `head`
3. ฟังก์ชัน `void addTerm(int coef, int exponent)` เพิ่มพจน์ใหม่ให้กับ `Polynomial`
4. ฟังก์ชัน `void plus(Polynomial f2)` รวม `Polynomial f2` เข้ากับ `Polynomial` ที่เรียก ตามคำอธิบายข้างต้น
5. ฟังก์ชัน `void minus(Polynomial f2)` ลบ `Polynomial f2` ออกจาก `Polynomial` ที่เรียก ตามคำอธิบายข้างต้น

ฟังก์ชันที่นักศึกษาต้องเขียนส่งคือ Polynomial(), addTerm(), plus() และ minus()

นักศึกษสามารถสร้างตัวแปรหรือฟังก์ชันเพิ่มได้ตามความเหมาะสม (ไม่ตรวจ)

6. ส่งไฟล์ HW03.cpp ที่ <https://grader.cs.science.cmu.ac.th> คะแนนที่ได้ในเว็บเกรดเดอร์คือคะแนนที่นำไปใช้ตัดเกรด

ตัวอย่างการรันโปรแกรมที่ถูกต้อง



```
1 #include <iostream>
2 #include "HW03.cpp"
3 using namespace std;
4
5 int main()
6 {
7     Polynomial f1;
8     Polynomial f2;
9     f1.addTerm(5,8);
10    f1.addTerm(4,3);
11    f1.addTerm(2,1);
12    f1.printPolynomial();
13    f2.addTerm(4,5);
14    f2.addTerm(1,1);
15    f2.addTerm(-5,0);
16    f2.printPolynomial();
17    f1.plus(f2);
18    f1.printPolynomial();
19    return 0;
20 }
```

```
> clang++-7 -pthread -std=c++17 -o main HW03 main.cpp term.cpp
> ./main
[ 5X^8 4X^3 2X^1 ]
[ 4X^5 1X^1 -5X^0 ]
[ 5X^8 4X^5 4X^3 3X^1 -5X^0 ]
```

Compiler Note:

- ถ้ารันผ่าน Command Line สามารถใช้ `g++ -o [ชื่อโปรแกรม] term.cpp HW03.cpp main.cpp`
- ถ้าใช้ VS Code สามารถดูตัวอย่างการตั้งค่าได้ที่

<https://code.visualstudio.com/docs/cpp/introvideos-cpp>