📌 Coding Challenge: Polynomial Addition Using Linked List

📝 Problem Statement

You are given two polynomials represented as **linked lists**.
Each node stores:

- `coefficient` (int)
- `power` (int)

The linked lists are in **decreasing order of power** (highest power first).

Your task is to **add the two polynomials** and return the resulting polynomial as a linked list in the same sorted order.

Each polynomial can have:

- positive/negative coefficients
- missing powers (meaning coefficient = 0 for that power)

If the sum of coefficients for a power becomes **0**, skip that term—don't include it in the result.

⚙️ Constraints

- 1≤Number of terms≤1041 ≤ Number\ of\ terms ≤ 10^41≤Number of terms≤104
- Powers are non-negative integers
- Linked list nodes are sorted in strictly decreasing order of power

---

✔️ Example 1

Input

```
Polynomial A: 5x^3 + 4x^2 + 2x^0
Polynomial B: 5x^3 + 5x^1 + 5x^0
```

Linked List Format

```
A = [(5,3) -> (4,2) -> (2,0)]
B = [(5,3) -> (5,1) -> (5,0)]
```

Output

```
Result: 10x^3 + 4x^2 + 5x^1 + 7x^0
```

Linked List Format

```
[(10,3) -> (4,2) -> (5,1) -> (7,0)]
```

✔️ Example 2

Input

```
Polynomial A: 3x^4 + 2x^2 + 7x^1
Polynomial B: 4x^3 + 2x^2 + 3x^1
```

Linked List Format

```
A = [(3,4) -> (2,2) -> (7,1)]
B = [(4,3) -> (2,2) -> (3,1)]
```

Output

```
Result: 3x^4 + 4x^3 + 4x^2 + 10x^1
```

Linked List Format

```
[(3,4) -> (4,3) -> (4,2) -> (10,1)]
```

✔️ Example 3 — Zero Terms Removal

Input

```
Polynomial A: 5x^2 + 3x^1
Polynomial B: -5x^2 + 10x^0
```

Output

```
Result: 3x^1 + 10x^0
```

📌 Your Task

Write a function:

```
Node addPolynomial(Node poly1, Node poly2)
```

Where each node contains:

```
int coeff;
int power;
Node next;
```

The function must return a **sorted linked list** representing the sum of the two polynomials.