# Problem 1634: Add Two Polynomials Represented as Linked Lists

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 60.89%
**Paid Only:** Yes
**Tags:** Linked List, Math, Two Pointers

## Problem Description

A polynomial linked list is a special type of linked list where every node represents a term in a polynomial expression.

Each node has three attributes:

* `coefficient`: an integer representing the number multiplier of the term. The coefficient of the term `**9** x4` is `9`. * `power`: an integer representing the exponent. The power of the term `9x**4**` is `4`. * `next`: a pointer to the next node in the list, or `null` if it is the last node of the list.

For example, the polynomial `5x3 + 4x - 7` is represented by the polynomial linked list illustrated below:

![](https://assets.leetcode.com/uploads/2020/09/30/polynomial2.png)

The polynomial linked list must be in its standard form: the polynomial must be in **strictly** descending order by its `power` value. Also, terms with a `coefficient` of `0` are omitted.

Given two polynomial linked list heads, `poly1` and `poly2`, add the polynomials together and return _the head of the sum of the polynomials_.

**`PolyNode` format:**

The input/output format is as a list of `n` nodes, where each node is represented as its `[coefficient, power]`. For example, the polynomial `5x3 + 4x - 7` would be represented as:

`[[5,3],[4,1],[-7,0]]`.

**Example 1:**

![](https://assets.leetcode.com/uploads/2020/10/14/ex1.png)

**Input:** poly1 = [[1,1]], poly2 = [[1,0]] **Output:** [[1,1],[1,0]] **Explanation:** poly1 = x. poly2 = 1. The sum is x + 1.

**Example 2:**

**Input:** poly1 = [[2,2],[4,1],[3,0]], poly2 = [[3,2],[-4,1],[-1,0]] **Output:** [[5,2],[2,0]] **Explanation:** poly1 = 2x2 + 4x + 3. poly2 = 3x2 - 4x - 1. The sum is 5x2 + 2. Notice that we omit the "0x" term.

**Example 3:**

**Input:** poly1 = [[1,2]], poly2 = [[-1,2]] **Output:** [] **Explanation:** The sum is 0. We return an empty list.

**Constraints:**

* `0 <= n <= 104` * `-109 <= PolyNode.coefficient <= 109` * `PolyNode.coefficient != 0` * `0 <= PolyNode.power <= 109` * `PolyNode.power > PolyNode.next.power`

## Code Snippets

**C++:**

```
/**
* Definition for polynomial singly-linked list.
* struct PolyNode {
* int coefficient, power;
* PolyNode *next;
* PolyNode(): coefficient(0), power(0), next(nullptr) {};
* PolyNode(int x, int y): coefficient(x), power(y), next(nullptr) {};
* PolyNode(int x, int y, PolyNode* next): coefficient(x), power(y),
next(next) {};
* };
```

```
*/

class Solution {
public:
PolyNode* addPoly(PolyNode* poly1, PolyNode* poly2) {

}
};
```

### Java:

```
/**
* Definition for polynomial singly-linked list.
* class PolyNode {
* int coefficient, power;
* PolyNode next = null;

* PolyNode() {}
* PolyNode(int x, int y) { this.coefficient = x; this.power = y; }
* PolyNode(int x, int y, PolyNode next) { this.coefficient = x; this.power =
y; this.next = next; }
* }
*/

class Solution {
public PolyNode addPoly(PolyNode poly1, PolyNode poly2) {

}
}
```

### Python3:

```
# Definition for polynomial singly-linked list.
# class PolyNode:
# def __init__(self, x=0, y=0, next=None):
# self.coefficient = x
# self.power = y
# self.next = next

class Solution:
def addPoly(self, poly1: 'PolyNode', poly2: 'PolyNode') -> 'PolyNode':
```