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) {

    }
}
```

JavaScript

```
/**
 * Definition for polynomial singly-linked list.
 * function PolyNode(x=0, y=0, next=null) {
 *     this.coefficient = x;
 *     this.power = y;
 *     this.next = next;
 * }
 */
```

```javascript
/**
 * @param {PolyNode} poly1
 * @param {PolyNode} poly2
 * @return {PolyNode}
 */
var addPoly = function(poly1, poly2) {

};
```

C++

```cpp
/**
 * 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) {

    }
};
```

C#

```
/**
 * Definition for polynomial singly-linked list.
 * public class PolyNode {
 *     public int coefficient, power;
 *     public PolyNode next;
 *
 *     public PolyNode(int x=0, int y=0, PolyNode next=null) {
 *         this.coefficient = x;
 *         this.power = y;
 *         this.next = next;
 *     }
 * }
 */

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

    }
}
```
