class Node:

    def \_\_init\_\_(self, coeff, power):

        self.coeff = coeff

        self.power = power

[self.next](http://self.next/) = None

class Polynomial:

    def \_\_init\_\_(self):

        self.head = None

    def append(self, coeff, power):

        new\_node = Node(coeff, power)

        if self.head is None:

            self.head = new\_node

            return

        # Insert at the end

        current = self.head

        while [current.next](http://current.next/):

            current = [current.next](http://current.next/)

[current.next](http://current.next/) = new\_node

    def print\_poly(self):

        terms = []

        current = self.head

        while current:

            c = current.coeff

            p = current.power

            if c != 0:

                if p == 0:

                    terms.append(f"{c}")

                elif p == 1:

                    terms.append(f"{c}x")

                else:

                    terms.append(f"{c}x^{p}")

            current = [current.next](http://current.next/)

        if not terms:

            print("0")

        else:

            print(" + ".join(terms))

    def add(self, other):

        p1 = self.head

        p2 = other.head

        result = Polynomial()

        # Traverse both lists and add terms

        while p1 and p2:

            if p1.power == p2.power:

                sum\_coeff = p1.coeff + p2.coeff

                if sum\_coeff != 0:

                    result.append(sum\_coeff, p1.power)

                p1 = [p1.next](http://p1.next/)

                p2 = [p2.next](http://p2.next/)

            elif p1.power > p2.power:

                result.append(p1.coeff, p1.power)

                p1 = [p1.next](http://p1.next/)

            else:

                result.append(p2.coeff, p2.power)

                p2 = [p2.next](http://p2.next/)

        # Append remaining terms

        while p1:

            result.append(p1.coeff, p1.power)

            p1 = [p1.next](http://p1.next/)

        while p2:

            result.append(p2.coeff, p2.power)

            p2 = [p2.next](http://p2.next/)

        return result

# Example usage

poly1 = Polynomial()

poly1.append(3, 4)  # 3x^4

poly1.append(2, 3)  # 2x^3

poly1.append(1, 0)  # 1

poly2 = Polynomial()

poly2.append(5, 3)  # 5x^3

poly2.append(2, 1)  # 2x

poly2.append(3, 0)  # 3

print("Polynomial 1:")

poly1.print\_poly()

print("Polynomial 2:")

poly2.print\_poly()

result = poly1.add(poly2)

print("Sum:")

result.print\_poly()