

# Rajalakshmi Engineering College

Name: Tarrun B  
Email: 241801291@rajalakshmi.edu.in  
Roll no: 241801291  
Phone: 9840381059  
Branch: REC  
Department: I AI & DS FD  
Batch: 2028  
Degree: B.E - AI & DS

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 1\_COD\_Question 1

Attempt : 2  
Total Mark : 10  
Marks Obtained : 8

#### Section 1 : Coding

##### 1. Problem Statement

Janani is a tech enthusiast who loves working with polynomials. She wants to create a program that can add polynomial coefficients and provide the sum of their coefficients.

The polynomials will be represented as a linked list, where each node of the linked list contains a coefficient and an exponent. The polynomial is represented in the standard form with descending order of exponents.

##### *Input Format*

The first line of input consists of an integer  $n$ , representing the number of terms in the first polynomial.

The following  $n$  lines of input consist of two integers each: the coefficient and the exponent of the term in the first polynomial.

The next line of input consists of an integer  $m$ , representing the number of terms in the second polynomial.

The following  $m$  lines of input consist of two integers each: the coefficient and the exponent of the term in the second polynomial.

### **Output Format**

The output prints the sum of the coefficients of the polynomials.

### **Sample Test Case**

Input: 3

2 2

3 1

4 0

3

2 2

3 1

4 0

Output: 18

### **Answer**

# You are using Python

class Node:

```
def __init__(self, coefficient, exponent):
```

```
    self.coefficient=coefficient
```

```
    self.exponent=exponent
```

```
    self.next=None
```

class Polynomial:

```
def __init__(self):
```

```
    self.head=None
```

```
def insert_term(self, coefficient, exponent):
```

```
    new_node=Node(coefficient,exponent)
```

```
    if self.head is None or self.head.exponent < exponent:
```

```
        new_node.next=self.head
```

```
        self.head=new_node
```

```
        return
```

```
    temp=self.head
```

```
    while temp.next and temp.next.exponent > exponent:
```

```

    if temp.next.exponent == exponent:
        temp.next.coefficient += coefficient
        return
    temp = temp.next
    if temp.exponent == exponent:
        temp.coefficient += coefficient

    else:
        new_node.next = temp.next
        temp.next = new_node
def sum_of_coefficients(self):
    temp = self.head
    total = 0
    while temp:
        total += temp.coefficient
        temp = temp.next
    return total
@staticmethod
def add_polynomials(poly1, poly2):
    result = Polynomial()
    temp1, temp2 = poly1.head, poly2.head

    while temp1 and temp2:
        if temp1.exponent > temp2.exponent:
            result.insert_term(temp1.coefficient, temp1.exponent)
            temp1 = temp1.next
        elif temp1.exponent < temp2.exponent:
            result.insert_term(temp2.coefficient, temp2.exponent)
            temp2 = temp2.next
        else:
            result.insert_term(temp1.coefficient
+temp2.coefficient, temp1.exponent)
            temp1 = temp1.next
            temp2 = temp2.next
    while temp1:
        result.insert_term(temp1.coefficient, temp1.exponent)
        temp1 = temp1.next
    while temp2:
        result.insert_term(temp2.coefficient, temp2.exponent)
        temp2 = temp2.next
    return result

```

```
n=int(input().strip())
poly1=Polynomial()
for _ in range(n):
    coef,exp = map(int,input().strip().split())
    poly1.insert_term(coef, exp)
m=int(input().strip())
poly2=Polynomial()
for _ in range(m):
    coef,exp = map(int,input().strip().split())
    poly2.insert_term(coef, exp)
result_poly=Polynomial.add_polynomials(poly1,poly2)
print(result_poly.sum_of_coefficients())
```

**Status :** Partially correct

**Marks :** 8/10