

# Rajalakshmi Engineering College

Name: janane jaipratha  
Email: 241501072@rajalakshmi.edu.in  
Roll no: 241501072  
Phone: 7548851756  
Branch: REC  
Department: I AIML AD  
Batch: 2028  
Degree: B.E - AI & ML

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 : 10

#### 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**

```
#include<stdio.h>
#include<stdlib.h>
struct Node{
    int coef;
    int exp;
    struct Node* next;
};
struct Node* createNode(int coef,int exp){
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->coef = coef;
    newNode->exp = exp;
    newNode->next = NULL;
    return newNode;
}
void insertNode(struct Node** head,int coef,int exp){
    struct Node* newNode = createNode(coef,exp);
    if(*head == NULL){
        *head = newNode;
    }else{
```

```

    struct Node* temp = *head;
    while(temp->next != NULL){
        temp = temp->next;
    }
    temp->next = newNode;
}
}

struct Node* addPolynomials(struct Node* poly1, struct Node* poly2){
    struct Node* result = NULL;
    struct Node *p1 = poly1, *p2 = poly2;
    while(p1 != NULL || p2 != NULL){
        if(p1 != NULL && (p2 == NULL || p1->exp > p2->exp)){
            insertNode(&result, p1->coef, p1->exp);
            p1 = p1->next;
        } else if(p2 != NULL && (p1 == NULL || p2->exp > p1->exp)){
            insertNode(&result, p2->coef, p2->exp);
            p2 = p2->next;
        } else{
            int sumCoef = p1->coef + p2->coef;
            if(sumCoef != 0){
                insertNode(&result, sumCoef, p1->exp);
            }
            p1 = p1->next;
            p2 = p2->next;
        }
    }
    return result;
}

int sumCoefficients(struct Node* head){
    int sum = 0;
    struct Node* temp = head;
    while(temp != NULL){
        sum += temp->coef;
        temp = temp->next;
    }
    return sum;
}

void freePolynomial(struct Node* head){
    struct Node* temp;
    while(head != NULL){
        temp = head;
        head = head->next;
    }
}

```

```

    free(temp);
}
}
int main(){
    int n,m;
    struct Node *poly1 = NULL, *poly2 = NULL;
    scanf("%d",&n);
    for(int i = 0;i < n;i++){
        int coef,exp;
        scanf("%d %d",&coef,&exp);
        insertNode(&poly1,coef,exp);
    }
    scanf("%d",&m);
    for(int i = 0;i < m;i++){
        int coef,exp;
        scanf("%d %d",&coef,&exp);
        insertNode(&poly2,coef,exp);
    }
    struct Node* result = addPolynomials(poly1,poly2);
    printf("%d\n",sumCoefficients(result));
    freePolynomial(poly1);
    freePolynomial(poly2);
    freePolynomial(result);
    return 0;
}

```

**Status :** Correct

**Marks :** 10/10