Rajalakshmi Engineering College

Name: vinnarasan j
Email: 241501245@rajalakshmi.edu.in
Roll no: 241501245
Phone: 7305999373
Branch: REC
Department: I AI & ML FC
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

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>

// Structure for a node in the linked list representing a term in the polynomial
struct Node {
    int coefficient;
    int exponent;
    struct Node* next;
};

// Function to insert a new term into the polynomial linked list (maintaining
descending order of exponents)
struct Node* insertTerm(struct Node* head, int coeff, int exp) {
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
    if (new_node == NULL) {
        printf("Memory allocation failed\n");
        exit(EXIT_FAILURE);
    }
    new_node->coefficient = coeff;
```

```
    new_node->exponent = exp;
    new_node->next = NULL;

    if (head == NULL || exp > head->exponent) {
        new_node->next = head;
        return new_node;
    }

    struct Node* current = head;
    while (current->next != NULL && exp < current->next->exponent) {
        current = current->next;
    }
    new_node->next = current->next;
    current->next = new_node;
    return head;
}

// Function to add two polynomial linked lists
struct Node* addPolynomials(struct Node* poly1, struct Node* poly2) {
    struct Node* result = NULL;
    struct Node* current1 = poly1;
    struct Node* current2 = poly2;

    while (current1 != NULL || current2 != NULL) {
        int coeff = 0;
        int exp;

        if (current1 != NULL && current2 != NULL && current1->exponent == current2->exponent) {
            coeff = current1->coefficient + current2->coefficient;
            exp = current1->exponent;
            current1 = current1->next;
            current2 = current2->next;
        } else if (current1 != NULL && (current2 == NULL || current1->exponent > current2->exponent)) {
            coeff = current1->coefficient;
            exp = current1->exponent;
            current1 = current1->next;
        } else if (current2 != NULL && (current1 == NULL || current2->exponent > current1->exponent)) {
            coeff = current2->coefficient;
            exp = current2->exponent;
```

```
            current2 = current2->next;
        }

        if (coeff != 0) {
            result = insertTerm(result, coeff, exp);
        }
    }
    return result;
}

// Function to calculate the sum of coefficients of a polynomial
int sumOfCoefficients(struct Node* poly) {
    int sum = 0;
    struct Node* current = poly;
    while (current != NULL) {
        sum += current->coefficient;
        current = current->next;
    }
    return sum;
}

// Function to free the memory allocated for the linked list
void freePolynomial(struct Node* head) {
    struct Node* current = head;
    struct Node* next;
    while (current != NULL) {
        next = current->next;
        free(current);
        current = next;
    }
}

int main() {
    int n, m, coeff, exp;
    struct Node* poly1 = NULL;
    struct Node* poly2 = NULL;
    struct Node* sum_poly = NULL;

    // Read the first polynomial
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        scanf("%d %d", &coeff, &exp);
```

```
        poly1 = insertTerm(poly1, coeff, exp);
    }

    // Read the second polynomial
    scanf("%d", &m);
    for (int i = 0; i < m; i++) {
        scanf("%d %d", &coeff, &exp);
        poly2 = insertTerm(poly2, coeff, exp);
    }

    // Add the two polynomials
    sum_poly = addPolynomials(poly1, poly2);

    // Calculate and print the sum of coefficients of the resulting polynomial
    printf("%d\n", sumOfCoefficients(sum_poly));

    // Free the allocated memory
    freePolynomial(poly1);
    freePolynomial(poly2);
    freePolynomial(sum_poly);

    return 0;
}
```

Status : Correct **Marks : 10/10**