

Rajalakshmi Engineering College

Name: NITHIN S P
Email: 240801229@rajalakshmi.edu.in
Roll no: 240801229
Phone: 6369443065
Branch: REC
Department: I ECE AF
Batch: 2028
Degree: B.E - ECE

Scan to verify results



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 1

Attempt : 1
Total Mark : 1
Marks Obtained : 1

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>
```

```
typedef struct Polynomial{
    int coefficient;
    int exponential;
    struct Polynomial* next;
}Node;
```

```
Node* newnode(int coefficient,int exponential){
    Node*new_node=(Node*)malloc(sizeof(Node));
    new_node->coefficient = coefficient;
    new_node->exponential = exponential;
    new_node->next=NULL;
    return new_node;
}
```

```
Node*input(int n){
    int c,e;
    scanf("%d %d",&c,&e);
```

```

Node* poly=newnode(c,e);
Node* ptr=poly;

for(int i=1;i<n;i++){
    scanf("%d%d",&c,&e);
    ptr->next=newnode(c,e);
    ptr=ptr->next;
}
return poly;
}

int csum(Node* poly){
    int sum=0;
    Node* ptr=poly;
    while(ptr){
        sum+=ptr->coefficient;
        ptr=ptr->next;
    }
    return sum;
}

int main(){
    int sum=0;
    int n;
    scanf("%d",&n);
    Node* poly1=input(n);
    scanf("%d",&n);
    Node* poly2=input(n);
    sum+=csum(poly1);
    sum+=csum(poly2);
    printf("%d",sum);
}

```

Status : Partially correct

Marks : 1/1