

Rajalakshmi Engineering College

Name: CHEMBETI JAHNAVI
Email: 240701089@rajalakshmi.edu.in
Roll no: 240701089
Phone: 6300874727
Branch: REC
Department: I CSE AG
Batch: 2028
Degree: B.E - CSE

Scan to verify results



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_week 1_CY

Attempt : 1
Total Mark : 30
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Keerthi is a tech enthusiast and is fascinated by polynomial expressions. She loves to perform various operations on polynomials.

Today, she is working on a program to multiply two polynomials and delete a specific term from the result.

Keerthi needs your help to implement this program. She wants to take the coefficients and exponents of the terms of the two polynomials as input, perform the multiplication, and then allow the user to specify an exponent for deletion from the resulting polynomial, and display the result.

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 representing the coefficient and the exponent of the term in the first polynomial.

The next line 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 representing the coefficient and the exponent of the term in the second polynomial.

The last line consists of an integer, representing the exponent of the term that Keerthi wants to delete from the multiplied polynomial.

Output Format

The first line of output displays the resulting polynomial after multiplication.

The second line displays the resulting polynomial after deleting the specified term.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3

2 2

3 1

4 0

2

1 2

2 1

2

Output: Result of the multiplication: $2x^4 + 7x^3 + 10x^2 + 8x$

Result after deleting the term: $2x^4 + 7x^3 + 8x$

Answer

```
// You are using GCC
```

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
typedef struct Polynomial {
```

```

    int coeff;
    int ex;
    struct Polynomial* next;
}Node;
Node* newnode(int coeff, int ex) {
    Node* Polynomial = (Node*)malloc(sizeof(Node));
    Polynomial->coeff = coeff;
    Polynomial->ex = ex;
    Polynomial->next = NULL;
    return Polynomial;
}
void insertNode(Node** head, int coeff, int ex){
    if(*head == NULL){
        *head = newnode(coeff, ex);
        return;
    }
    Node* temp = *head;
    while(temp->next != NULL){
        temp = temp->next;
    }
    temp->next = newnode(coeff, ex);
}
void traverse(Node* head){
    while(head->next!=NULL){
        printf("%dx^%d + ", head->coeff, head->ex);
        head=head->next;
    }
    printf("%dx^%d\n",head->coeff,head->ex);
}
int main() {
    int n, m, ex,coeff;
    Node* p1=NULL;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        scanf("%d %d",&coeff, &ex);
        insertNode(&p1,coeff,ex);
    }
    Node* p2=NULL;
    scanf("%d",&m);
    for(int i=0;i<m;i++){
        scanf("%d %d",&coeff,&ex);
        insertNode(&p2,coeff,ex);
    }
}

```

```

    }
    Node* temp1 = p1;
    while(temp1 != NULL){
        Node* temp2 = p2;
        while(temp2 != NULL){
            temp2->coeff=temp2->coeff*temp1->coeff;
            temp2->ex=temp2->ex*temp1->ex;
            temp2=temp2->next;
        }
        temp1=temp1->next;
    }
    traverse(p2);
}

```

Status : Wrong

Marks : 0/10

2. Problem Statement

Hasini is studying polynomials in her class. Her teacher has introduced a new concept of two polynomials using linked lists.

The teacher provides Hasini with a program that takes two polynomials as input, represented as linked lists, and then displays them together. The polynomials are simplified and should be displayed in the format ax^b , where a is the coefficient and b is the exponent.

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 first line of output prints the first polynomial.

The second line of output prints the second polynomial.

The polynomials should be displayed in the format ax^b , where a is the coefficient and b is the exponent.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3

1 2

2 1

3 0

3

2 2

1 1

4 0

Output: $1x^2 + 2x + 3$

$2x^2 + 1x + 4$

Answer

// You are using GCC

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
typedef struct Node{
```

```
    int coeff;
```

```
    int expo;
```

```
    struct Node* next;
```

```
}Node;
```

```
Node *createNode(int coeff,int expo){
```

```
    Node *newnode=(Node*)malloc(sizeof(Node));
```

```
    newnode->coeff=coeff;
```

```
    newnode->expo=expo;
```

```
    newnode->next=NULL;
```

```
    return newnode;
```

```
}
```

```
void insertTerm(Node **poly,int coeff,int expo){
```

```
    if(coeff==0){
```

```

    return;
}
Node *newnode=createNode(coeff,expo);
if(*poly==NULL||(*poly)->expo<expo){
    newnode->next= *poly;
    *poly=newnode;
    return;
}
Node *temp= *poly;
Node *prev=NULL;
while(temp && temp->expo>expo){
    prev=temp;
    temp=temp->next;
}
if(prev==NULL){
    newnode->next= *poly;
    *poly=newnode;
}
else{
    newnode->next=prev->next;
    prev->next=newnode;
}
}
void displayPolynomial(Node *poly){
    if(!poly){
        printf("0\n");
        return;
    }
    int first=1;
    while(poly){
        if(!first){
            if(poly->coeff>0){
                printf(" + ");
            }
            else{
                printf(" - ");
            }
        }else{
            if(poly->coeff<0){
                printf(" - ");
            }
        }
        first=0;
    }
}

```

```

    }
    if(poly->expo==0){
        printf("%d",abs(poly->coeff));
    }
    else if(poly->expo==1){
        printf("%dx",abs(poly->coeff));
    }else{
        printf("%dx^%d",abs(poly->coeff),poly->expo);
    }
    poly=poly->next;
}
printf("\n");
}
int main(){
    int n,m,coeff,expo;
    Node *poly1=NULL, *poly2=NULL;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        scanf("%d %d",&coeff,&expo);
        insertTerm(&poly1,coeff,expo);
    }
    scanf("%d",&m);
    for(int i=0;i<m;i++){
        scanf("%d %d",&coeff,&expo);
        insertTerm(&poly2,coeff,expo);
    }
    displayPolynomial(poly1);
    displayPolynomial(poly2);
    return 0;
}

```

Status : Correct

Marks : 10/10

3. Problem Statement

Timothy wants to evaluate polynomial expressions for his mathematics homework. He needs a program that allows him to input the coefficients of a polynomial based on its degree and compute the polynomial's value for a given input of x. Implement a function that takes the degree, coefficients, and the value of x, and returns the evaluated result of the polynomial.

Example

Input:

degree of the polynomial = 2

coefficient of x^2 = 13

coefficient of x^1 = 12

coefficient of x^0 = 11

$x = 1$

Output:

36

Explanation:

Calculate the value of $13x^2$: $13 * 1^2 = 13$.

Calculate the value of $12x^1$: $12 * 1^1 = 12$.

Calculate the value of $11x^0$: $11 * 1^0 = 11$.

Add the values of x^2 , x^1 , and x^0 together: $13 + 12 + 11 = 36$.

Input Format

The first line of input consists of an integer representing the degree of the polynomial.

The second line consists of an integer representing the coefficient of x^2 .

The third line consists of an integer representing the coefficient of x^1 .

The fourth line consists of an integer representing the coefficient of x^0 .

The fifth line consists of an integer representing the value of x , at which the polynomial should be evaluated.

Output Format

The output is an integer value obtained by evaluating the polynomial at the given value of x .

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2

13

12

11

1

Output: 36

Answer

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
typedef struct Poly{
    int x;
    int expon;
    struct Poly* next;
}Node;
Node* newnode(int x, int expon){
    Node* node = (Node*)malloc(sizeof(Node));
    node->x = x;
    node->expon = expon;
    node->next = NULL;
    return node;
}
void insertNode(Node** head, int x,int expon){
    Node* temp = *head;
    if(temp == NULL){
        *head = newnode(x, expon);
        return;
    }
    while(temp->next != NULL){
        temp = temp->next;
    }
    temp->next = newnode(x, expon);
}
int main(){
    int degree, x;
```

```
scanf("%d", &degree);
Node* head = NULL;
for(int i=0;i<=degree;i++){
    scanf("%d",&x);
    insertNode(&head, x,degree -i);
}
int value = 0;
int n;
scanf("%d", &n);
while(head != NULL){
    value +=head->x * pow(n, head->expon);
    head = head->next;
}
printf("%d", value);
}
```

Status : Correct

Marks : 10/10