



PUNE INSTITUTE OF COMPUTER TECHNOLOGY  
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No:22203

Submission date:

Title:

**Polynomial operation using linked list**

Problem  
Statement

Write a program in C to illustrate the addition of two polynomials using linked list. Test various input for addition, subtraction, and multiplication of two polynomial by following all math rules

**Programmer Name: Arpan Agrawal**

**Batch: E6**

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

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

```
// Structure to represent a node in the linked list
```

```
struct Node {
```

```
    int coeff;
```

```
    int exp;
```

```
    struct Node* next;
```

```
};
```

```
// Function to create a new node
```

```
struct Node* createNode(int coeff, int exp) {
```

```
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
```

```
    newNode->coeff = coeff;
```

```
    newNode->exp = exp;
```

```
    newNode->next = NULL;
```

```
    return newNode;
```

```
}
```

```
// Function to insert a new node in the polynomial list
```

```
void insertNode(struct Node** poly, int coeff, int exp) {
```

```
    struct Node* newNode = createNode(coeff, exp);
```

```
    if (*poly == NULL) {
```

```
        *poly = newNode;
```

```
    } else {
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY  
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No:22203

Submission date:

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

// Function to display a polynomial
void displayPolynomial(struct Node* poly) {
    while (poly != NULL) {
        printf("%dx^%d", poly->coeff, poly->exp);
        poly = poly->next;
        if (poly != NULL) {
            printf(" + ");
        }
    }
    printf("\n");
}

// Function to add two polynomials
struct Node* addPolynomials(struct Node* poly1, struct Node* poly2) {
    struct Node* result = NULL;
    while (poly1 != NULL && poly2 != NULL) {
        if (poly1->exp > poly2->exp) {
            insertNode(&result, poly1->coeff, poly1->exp);
            poly1 = poly1->next;
        } else if (poly1->exp < poly2->exp) {
            insertNode(&result, poly2->coeff, poly2->exp);
            poly2 = poly2->next;
        } else {
            insertNode(&result, poly1->coeff + poly2->coeff, poly1->exp);
            poly1 = poly1->next;
        }
    }
}
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY  
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No:22203

Submission date:

```
        poly2 = poly2->next;
    }
}

// If there are remaining terms in poly1 or poly2
while (poly1 != NULL) {
    insertNode(&result, poly1->coeff, poly1->exp);
    poly1 = poly1->next;
}
while (poly2 != NULL) {
    insertNode(&result, poly2->coeff, poly2->exp);
    poly2 = poly2->next;
}

return result;
}

// Function to subtract two polynomials
struct Node* subtractPolynomials(struct Node* poly1, struct Node* poly2) {
    struct Node* result = NULL;
    while (poly1 != NULL && poly2 != NULL) {
        if (poly1->exp > poly2->exp) {
            insertNode(&result, poly1->coeff, poly1->exp);
            poly1 = poly1->next;
        } else if (poly1->exp < poly2->exp) {
            insertNode(&result, -poly2->coeff, poly2->exp);
            poly2 = poly2->next;
        } else {
            insertNode(&result, poly1->coeff - poly2->coeff, poly1->exp);
            poly1 = poly1->next;
            poly2 = poly2->next;
        }
    }
}
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY  
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No:22203

Submission date:

```
// If there are remaining terms in poly1 or poly2
while (poly1 != NULL) {
    insertNode(&result, poly1->coeff, poly1->exp);
    poly1 = poly1->next;
}
while (poly2 != NULL) {
    insertNode(&result, -poly2->coeff, poly2->exp);
    poly2 = poly2->next;
}

return result;
}

// Function to multiply two polynomials
struct Node* multiplyPolynomials(struct Node* poly1, struct Node* poly2) {
    struct Node* result = NULL;
    struct Node* temp2 = poly2;

    while (poly1 != NULL) {
        while (poly2 != NULL) {
            insertNode(&result, poly1->coeff * poly2->coeff, poly1->exp + poly2->exp);
            poly2 = poly2->next;
        }
        poly2 = temp2;
        poly1 = poly1->next;
    }
    return result;
}

// Function to simplify the polynomial by combining terms with the same
exponent
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY  
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No:22203

Submission date:

```
struct Node* simplifyPolynomial(struct Node* poly) {
    struct Node* ptr1 = poly;
    struct Node* ptr2;
    struct Node* prev;

    while (ptr1 != NULL && ptr1->next != NULL) {
        prev = ptr1;
        ptr2 = ptr1->next;

        while (ptr2 != NULL) {
            if (ptr1->exp == ptr2->exp) {
                ptr1->coeff += ptr2->coeff;
                prev->next = ptr2->next;
                free(ptr2);
                ptr2 = prev->next;
            } else {
                prev = ptr2;
                ptr2 = ptr2->next;
            }
        }
        ptr1 = ptr1->next;
    }

    return poly;
}
```

```
// Function to input a polynomial from the user
struct Node* inputPolynomial() {
    struct Node* poly = NULL;
    int n, coeff, exp;

    printf("Enter the number of terms: ");
    scanf("%d", &n);
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY  
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No:22203

Submission date:

```
for (int i = 0; i < n; i++) {
    printf("Enter coefficient and exponent for term %d: ", i + 1);
    scanf("%d %d", &coeff, &exp);
    insertNode(&poly, coeff, exp);
}

return poly;
}

int main() {
    struct Node *poly1 = NULL, *poly2 = NULL, *result = NULL;

    printf("Input first polynomial:\n");
    poly1 = inputPolynomial();

    printf("Input second polynomial:\n");
    poly2 = inputPolynomial();

    printf("First Polynomial: ");
    displayPolynomial(poly1);
    printf("Second Polynomial: ");
    displayPolynomial(poly2);

    // Add Polynomials
    result = addPolynomials(poly1, poly2);
    printf("\nAdded Polynomial: ");
    displayPolynomial(result);

    // Subtract Polynomials
    result = subtractPolynomials(poly1, poly2);
    printf("\nSubtracted Polynomial: ");
    displayPolynomial(result);
}
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY  
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No:22203

Submission date:

```
// Multiply Polynomials
result = multiplyPolynomials(poly1, poly2);
result = simplifyPolynomial(result);
printf("\nMultiplied Polynomial: ");
displayPolynomial(result);

return 0;
}
```

## Output :-

Input first polynomial:

Enter the number of terms: 5.

Enter coefficient and exponent for term 1: 4 4

Enter coefficient and exponent for term 2: 3 3

Enter coefficient and exponent for term 3: 21 5

Enter coefficient and exponent for term 4: 9 6

Enter coefficient and exponent for term 5: 5 2

Input second polynomial:

Enter the number of terms: 2

Enter coefficient and exponent for term 1: 1 5

Enter coefficient and exponent for term 2: 5 6

First Polynomial:  $4x^4 + 3x^3 + 21x^5 + 9x^6 + 5x^2$

Second Polynomial:  $1x^5 + 5x^6$

Added Polynomial:  $1x^5 + 5x^6 + 4x^4 + 3x^3 + 21x^5 + 9x^6 + 5x^2$

Subtracted Polynomial:  $-1x^5 + -5x^6 + 4x^4 + 3x^3 + 21x^5 + 9x^6 + 5x^2$

Multiplied Polynomial:  $19x^9 + 41x^{10} + 28x^8 + 114x^{11} + 45x^{12} + 5x^7$