PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Autonomous Institute)

Class : SY BTech **Acad. Yr. 2025-26** **Semester : I**

Name of the student: Hariom Shrikrishna Gundale **PRN : 124B1B036**

Department: Computer Enginnering **Division : A**

Course Name : Data Structures Laboratory **Course Code: BCE23PC02**

Completion Date : 11/08/2025

Assignment No. 5

Problem Statement: Write a program to perform Polynomial Addition using Linked Lists

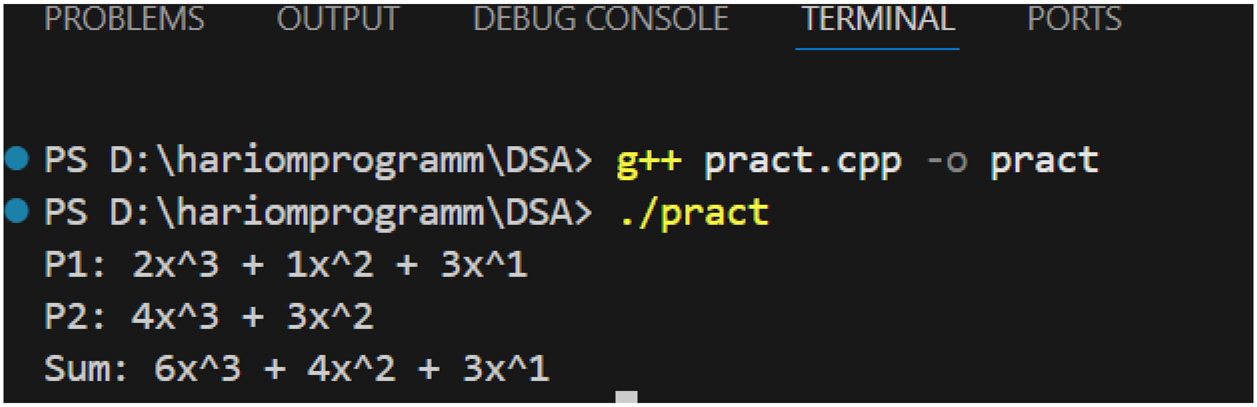
- Each term is a node (with coefficient and power).
- Add two polynomials represented by linked lists**.**

Source Code :

```
#include <iostream>
using namespace std;

class node {
public:
   int coeff;
   int pow;
   node* next;

   node(int c, int p) {
       coeff = c;
       pow = p;
       next = NULL;
   }
};

class ll {
   node* head;

public:
   ll() {
       head = NULL;
   }

   void create(int coeff, int pow) {
       node* nn = new node(coeff, pow);
       if (head == NULL) {
           head = nn;
       } else {
           node* temp = head;
           while (temp->next != NULL) {
               temp = temp->next;
           }
           temp->next = nn;
       }
   }

   void polyadd(ll& l1, ll& l2) {
       node* p1 = l1.head;
       node* p2 = l2.head;

       while (p1 != NULL && p2 != NULL) {
           if (p1->pow == p2->pow) {
               create(p1->coeff + p2->coeff, p1->pow);
               p1 = p1->next;
               p2 = p2->next;
           }
           else if (p1->pow > p2->pow) {
               create(p1->coeff, p1->pow);
               p1 = p1->next;
           }
           else {
               create(p2->coeff, p2->pow);
               p2 = p2->next;
           }
       }

       while (p1 != NULL) {
           create(p1->coeff, p1->pow);
           p1 = p1->next;
       }
       while (p2 != NULL) {
           create(p2->coeff, p2->pow);
           p2 = p2->next;
       }
   }

   void show() {
       node* p = head;
       if (!p) {
           cout << "Polynomial is empty" << endl;
           return;
       }
       while (p != NULL) {
           cout << p->coeff << "x^" << p->pow;
           p = p->next;
           if (p != NULL) cout << " + ";
       }
       cout << endl;
   }
};
```

```
int main() {
  ll l1, l2, l3;


  l1.create(2, 3);
  l1.create(1, 2);
  l1.create(3, 1);
  l2.create(4, 3);
  l2.create(3, 2);
  cout < < "P1: ";
  l1.show();
  cout < < "P2: ";
  l2.show();
  l3.polyadd(l1, l2);
  cout < < "Sum: ";
  l3.show();

  return 0;
}
```

Screen Shot of Output :



```
PS D:\hariomprogramm\DSA> g++ pract.cpp -o pract
PS D:\hariomprogramm\DSA> ./pract
P1: 2x^3 + 1x^2 + 3x^1
P2: 4x^3 + 3x^2
Sum: 6x^3 + 4x^2 + 3x^1
```

Conclusion:

Thus, we have successfully implemented the C++ program for Polynomial Addition using the Data Structure Linked List. Using above code we can add a two polynomial equation.