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: Varad Amol Pisale PRN : 124B1B043**

**Department: Computer Engineering Division : A**

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

**Completion Date : 13/08/2025**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assignment No.**

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>

#include <cstdlib>

using namespace std;

class node {

public:

    int power = 0;

    int coeff = 0;

    node\* next = NULL;

    node() {}

    node(int coef, int p) {

        coeff = coef;

        power = p;

    }

};

class LL {

public:

    node\* head = NULL;

    node\* tail = head;

    void insert\_var(int coef, int p) {

        node\* temp = new node(coef, p);

        if (head == NULL) {

            head = temp;

            tail = head;

        } else {

            tail->next = temp;

            tail = temp;

        }

    }

    void print() {

        if (!head) {

            cout << "Polynomial is empty.\n";

            return;

        }

        node\* it = head;

        while (it != NULL) {

            cout << it->coeff << "x^" << it->power;

            if (it->next != NULL) cout << " + ";

            it = it->next;

        }

        cout << "\n";

    }

    void add\_ply(LL poly1, LL poly2) {

        node\* temp1 = poly1.head;

        node\* temp2 = poly2.head;

        while (temp1 != NULL && temp2 != NULL) {

            if (temp1->power == temp2->power) {

                insert\_var(temp1->coeff + temp2->coeff, temp1->power);

                temp1 = temp1->next;

                temp2 = temp2->next;

            }

            else if (temp1->power > temp2->power) { // Descending order fix

                insert\_var(temp1->coeff, temp1->power);

                temp1 = temp1->next;

            }

            else {

                insert\_var(temp2->coeff, temp2->power);

                temp2 = temp2->next;

            }

        }

        while (temp1 != NULL) {

            insert\_var(temp1->coeff, temp1->power);

            temp1 = temp1->next;

        }

        while (temp2 != NULL) {

            insert\_var(temp2->coeff, temp2->power);

            temp2 = temp2->next;

        }

    }

    void clear() {

        node\* temp = head;

        while (temp) {

            node\* del = temp;

            temp = temp->next;

            delete del;

        }

        head = tail = NULL;

    }

};

int main() {

    LL poly1, poly2, sum;

    int choice;

    while (true) {

        cout << "\n--- Polynomial Menu ---\n";

        cout << "1. Insert term in Polynomial 1\n";

        cout << "2. Insert term in Polynomial 2\n";

        cout << "3. Display Polynomial 1\n";

        cout << "4. Display Polynomial 2\n";

        cout << "5. Add Polynomials\n";

        cout << "6. Exit\n";

        cout << "Enter choice: ";

        cin >> choice;

        if (choice == 1) {

            int c, p;

            cout << "Enter coefficient and power: ";

            cin >> c >> p;

            poly1.insert\_var(c, p);

        }

        else if (choice == 2) {

            int c, p;

            cout << "Enter coefficient and power: ";

            cin >> c >> p;

            poly2.insert\_var(c, p);

        }

        else if (choice == 3) {

            cout << "Polynomial 1: ";

            poly1.print();

        }

        else if (choice == 4) {

            cout << "Polynomial 2: ";

            poly2.print();

        }

        else if (choice == 5) {

            sum.clear();

            sum.add\_ply(poly1, poly2);

            cout << "Sum: ";

            sum.print();

        }

        else if (choice == 6) {

            break;

        }

        else {

            cout << "Invalid choice!\n";

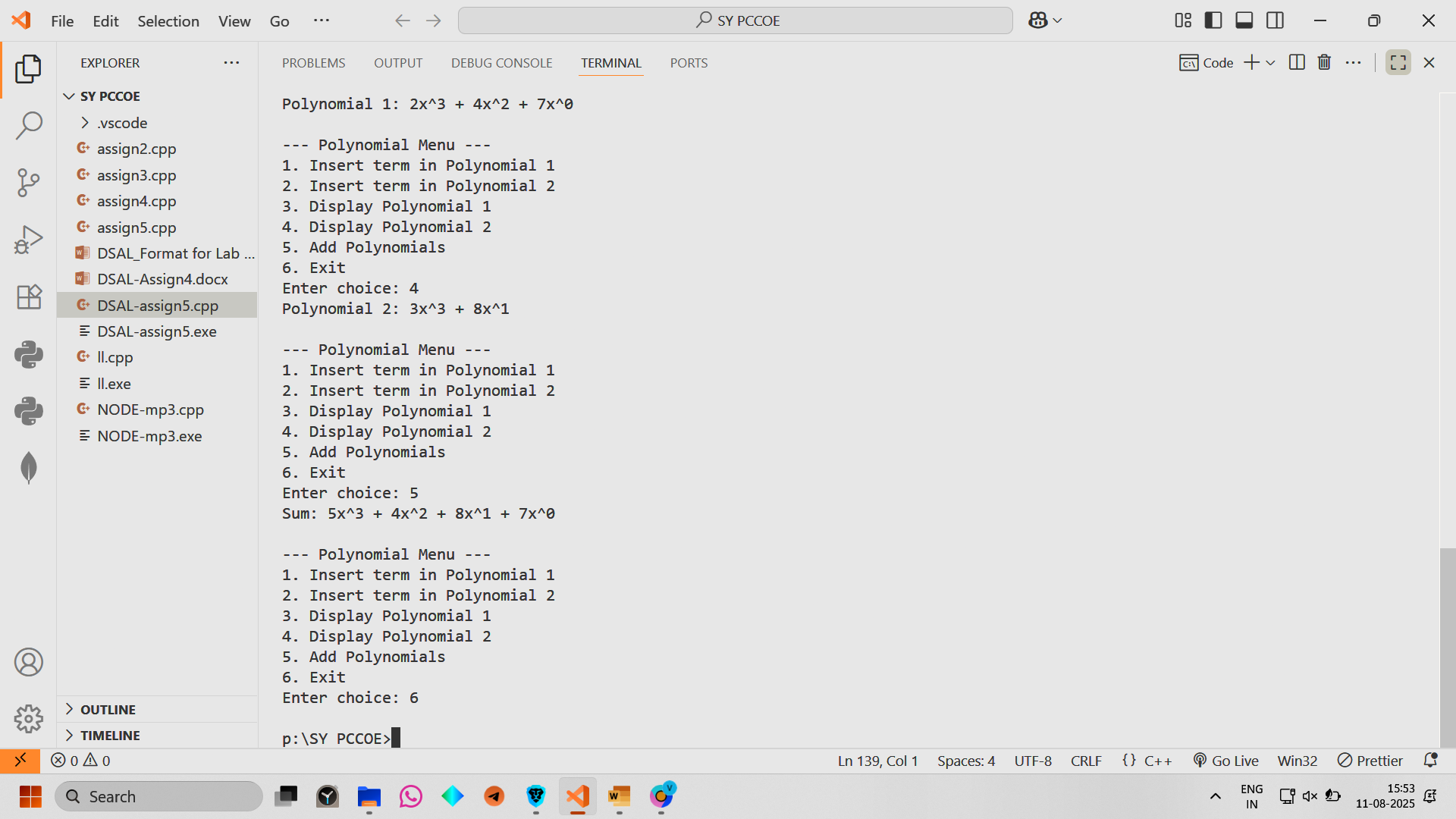
        }

    }

    return 0;

}

Screen Shot of Output :



Conclusion: Thus we have performed polynomial addition using Linked Lists