

NAME : PREM SAH
PRN : 123B1B234
DIV : D

ASSIGNMENT NO.6

Consider two polynomial expressions of any degree. Design solution to perform addition of these two polynomials with suitable data structure and display results.

CODE :

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

class Node {
public:
    int coef, exp;
    Node* next;
    Node(int c, int e) : coef(c), exp(e), next(nullptr) {}
};

class Polynomial {
private:
    Node* head;
public:
    Polynomial() : head(nullptr) {}

    void insert(int coef, int exp) {
        Node* newNode = new Node(coef, exp);
        if (!head || head->exp < exp) {
            newNode->next = head;
            head = newNode;
        } else {
            Node* temp = head;
            while (temp->next && temp->next->exp > exp) temp = temp->next;
            if (temp->exp == exp) temp->coef += coef;
            else {
                newNode->next = temp->next;
                temp->next = newNode;
            }
        }
    }

    Polynomial add(Polynomial& p) {
        Polynomial result;
        Node* p1 = head, *p2 = p.head;
        while (p1 || p2) {
            if (p1 && (!p2 || p1->exp > p2->exp)) {
                result.insert(p1->coef, p1->exp);
                p1 = p1->next;
            }
        }
    }
}
```

```

    } else if (p2 && (!p1 || p2->exp > p1->exp)) {
        result.insert(p2->coef, p2->exp);
        p2 = p2->next;
    } else {
        result.insert(p1->coef + p2->coef, p1->exp);
        p1 = p1->next;
        p2 = p2->next;
    }
}
return result;
}

void display() {
    Node* temp = head;
    while (temp) {
        cout << temp->coef << "x^" << temp->exp;
        if (temp->next) cout << " + ";
        temp = temp->next;
    }
    cout << endl;
}

};

int main() {
    Polynomial poly1, poly2;
    poly1.insert(3, 4); poly1.insert(2, 2); poly1.insert(5, 0);
    poly2.insert(4, 3); poly2.insert(1, 2); poly2.insert(7, 0);

    cout << "Poly 1: "; poly1.display();
    cout << "Poly 2: "; poly2.display();

    Polynomial result = poly1.add(poly2);
    cout << "Sum: "; result.display();

    return 0;
}

```

Output :

```

Poly 1: 3x^4 + 2x^2 + 5x^0
Poly 2: 4x^3 + 1x^2 + 7x^0
Sum: 3x^4 + 4x^3 + 3x^2 + 12x^0

```

```

main.cpp
1 #include <iostream>
2 using namespace std;
3 class Node {
4 public:
5     int coef, exp;
6     Node* next;
7     Node(int c, int e) : coef(c), exp(e), next(nullptr) {}
8 };
9 class Polynomial {
10 private:
11     Node* head;
12 public:
13     Polynomial() : head(nullptr) {}
14
15     void insert(int coef, int exp) {
16         Node* newNode = new Node(coef, exp);
17         if (!head || head->exp < exp) {
18             newNode->next = head;
19             head = newNode;
20         } else {
21             Node* temp = head;
22             while (temp->next && temp->next->exp > exp) temp = temp->next;
23             if (temp->next->exp == exp) temp->coef += coef;
24             else {
25                 newNode->next = temp->next;
26                 temp->next = newNode;
27             }
28         }
29     }
30
31     Polynomial add(Polynomial& p) {
32         Polynomial result;
33         Node* p1 = head, *p2 = p.head;
34         while (p1 || p2) {
35             if (p1 && (!p2 || p1->exp > p2->exp)) {
36                 result.insert(p1->coef, p1->exp);
37                 p1 = p1->next;
38             } else if (p2 && (!p1 || p2->exp > p1->exp)) {
39                 result.insert(p2->coef, p2->exp);
40                 p2 = p2->next;
41             } else {
42                 result.insert(p1->coef + p2->coef, p1->exp);
43                 p1 = p1->next;
44                 p2 = p2->next;
45             }
46         }
47         return result;
48     }
49
50     void display() {
51         Node* temp = head;
52         while (temp) {
53             cout << temp->coef << "x^" << temp->exp;
54             if (temp->next) cout << " + ";
55             temp = temp->next;
56         }
57         cout << endl;
58     }
59 };
60
61 int main() {
62     Polynomial poly1, poly2;
63     poly1.insert(3, 4); poly1.insert(2, 2); poly1.insert(5, 0);
64     poly2.insert(4, 3); poly2.insert(1, 2); poly2.insert(7, 0);
65     cout << "Poly 1: "; poly1.display();
66     cout << "Poly 2: "; poly2.display();
67     Polynomial result = poly1.add(poly2);
68     cout << "Sum: "; result.display();
69     return 0;
70 }

```

Output

/tmp/Zn7WtJ8nWE.o

Poly 1: 3x⁴ + 2x² + 5x⁰

Poly 2: 4x³ + 1x² + 7x⁰

Sum: 3x⁴ + 4x³ + 3x² + 12x⁰

=== Code Execution Successful ===