

Code คศธรรม เจริญธรรมกิจ 65090500415

```
#include <bits/stdc++.h>
using namespace std;
```

```
struct node
{
    int coeff;
    int exp;
    node *next;
};
```

```
class Polynomial
{
    node *head;
    node *tail;
    int size;
```

```
public:
    Polynomial()
    {
        head = NULL;
        tail = NULL;
        size = 0;
    }
```

```
void insert(int coeff, int exp)
{
    node *temp = new node;
    temp->coeff = coeff;
    temp->exp = exp;
    temp->next = NULL;
    if (head == NULL)
    {
        head = temp;
        tail = temp;
    }
    else
    {
        tail->next = temp;
        tail = temp;
    }
    size++;
}
```

```
}
```

```
void print_forward()
{
    node *temp = head;
    while (temp != NULL)
    {
        cout << temp->coeff << "x^" << temp->exp << " ";
        temp = temp->next;
    }
    cout << endl;
}
```

```
void print_backward()
{
    node *temp = tail;
    while (temp != NULL)
    {
        cout << temp->coeff << "x^" << temp->exp << " ";
        temp = temp->next;
    }
    cout << endl;
}
```

```
Polynomial add(Polynomial p)
{
    Polynomial result;
    node *temp1 = head;
    node *temp2 = p.head;
    while (temp1 != NULL && temp2 != NULL)
    {
        if (temp1->exp == temp2->exp)
        {
            result.insert(temp1->coeff + temp2->coeff, temp1->exp);
            temp1 = temp1->next;
            temp2 = temp2->next;
        }
        else if (temp1->exp > temp2->exp)
        {
            result.insert(temp1->coeff, temp1->exp);
            temp1 = temp1->next;
        }
        else
        {

```

```

        result.insert(temp2->coeff, temp2->exp);
        temp2 = temp2->next;
    }
}
while (temp1 != NULL)
{
    result.insert(temp1->coeff, temp1->exp);
    temp1 = temp1->next;
}
while (temp2 != NULL)
{
    result.insert(temp2->coeff, temp2->exp);
    temp2 = temp2->next;
}
return result;
}

```

Polynomial multiply(Polynomial p)

```

{
    Polynomial result;
    node *temp1 = head;
    node *temp2 = p.head;
    while (temp1 != NULL)
    {
        while (temp2 != NULL)
        {
            result.insert(temp1->coeff * temp2->coeff, temp1->exp + temp2->exp);
            temp2 = temp2->next;
        }
        temp1 = temp1->next;
        temp2 = p.head;
    }
    return result;
}
};

```

int main()

```

{
    Polynomial p1, p2, p3, p4;
    p1.insert(2, 3), p1.insert(3, 2), p1.insert(4, 1), p1.insert(5, 0);
    p2.insert(1, 2), p2.insert(2, 1), p2.insert(3, 0);
    p3 = p1.add(p2);
    p4 = p1.multiply(p2);
    cout << "A(x) = ";
}

```

```
p1.print_forward();  
cout << "B(x) = ";  
p2.print_forward();  
cout << "A(x) + B(x) = ";  
p3.print_forward();  
cout << "A(x) * B(x) = ";  
p4.print_forward();  
return 0;  
}
```

## Output

```
A(x) = 2x^3 3x^2 4x^1 5x^0  
B(x) = 1x^2 2x^1 3x^0  
A(x) + B(x) = 2x^3 4x^2 6x^1 8x^0  
A(x) * B(x) = 2x^5 4x^4 6x^3 3x^4 6x^3 9x^2 4x^3 8x^2 12x^1 5x^2 10x^1 15x^0
```