



PIMPRI CHINCHWAD EDUCATION TRUST'S.
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
(An Autonomous Institute)

S.Y. B. TECH

Name: Sonawane Prachi Mahendra.

Department : Computer Engineering

Course : Data Structures Laboratory

Date: 18/09/24

Year: 2024 – 25

Semester: I

PRN: 124B2B018

Division: B

Course Code: BCE23PC02

Assignment – 6

- **Aim:**

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

- **Source Code:**

```
#include<iostream>
```

```
class Node{
```

```
    public:
```

```
    int coeff; int exp;
```

```
    Node *next;
```

```
    Node()
```

```
    {
```

```
        coeff=0; exp=0;
```

```
        next=NULL;
```

```
    }
```

```
};
```

```

class LinkedList{ Node
*head; public:
LinkedList()
{
    head=NULL;
}

public: void insert(int value1,int value2)
{
    Node *nn=new Node();
    nn->coeff=value1;
    nn->exp=value2;
    if(head==NULL)
    {
        head=nn;
    }
    else
    {
        Node *temp=head;
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=nn;
    }
}

```

```

}

void add_Poly(LinkedList l1,LinkedList l2)
{
    Node *head1=l1.head; Node
    *head2=l2.head; Node *result=new
    Node(); Node *curr=result;

    while(head1!=NULL && head2!=NULL){ Node
    *nn=new Node();
    if(head1->exp>head2->exp)
    {
        nn->exp=head1->exp;
        nn->coeff=head1->coeff;
        head1=head1->next;
    }
    else if(head1->exp<head2->exp)
    {
        nn->exp=head2->exp;
        nn->coeff=head2->coeff;
        head2=head2->next;
    }else
    {
        nn->coeff=head1->coeff+head2->coeff;
        nn->exp=head1->exp;
    }
}

```

```

        head1=head1->next;

        head2=head2->next;

    }

    curr->next=nn;

    curr=curr->next;

}

curr->next=(head1!=NULL)?head1:head2;

head=result->next;

}

public: void display()

{

    Node *temp=head;

    while(temp->next!=NULL)

    {

        std::cout<<temp->coeff<<"^"<<temp->exp<<" ";

        temp=temp->next;

    }

    std::cout<<temp->coeff<<"^"<<temp->exp<<" "<<"\n";

}

};

int main()

{

    LinkedList l1,l2,l3; l1.insert(2,4);

    l1.insert(3,0);

```

```
l2.insert(9,3);  
l2.insert(7,0);  
l1.display();  
l2.display();  
l3.add_Poly(l1,l2);  
l3.display();  
}
```

- **Screen shots of Output:**

1.

```
Output
/tmp/qylipFUe0n.o
2^4 3^0
9^3 7^0
2^4 9^3 10^0

=== Code Execution Successful ===
```

- **Conclusion:**



Hence, we studied about how to perform operations on polynomial using linked list.