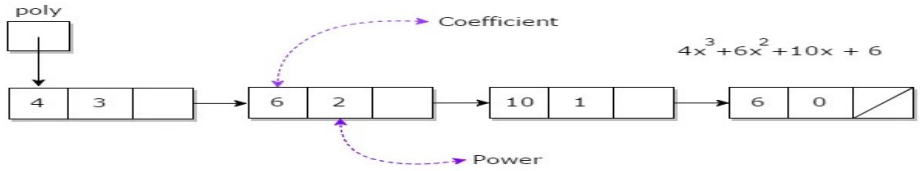
Experiment no: 6 Date:

Aim: Write a program to implement polynomial addition using singly linked list.

Programming Language: C/C++/Java

Theory: Linked list is a data structure that stores each element as an object in a node of the list. every node contains two data parts and links to the next node.Polynomial is a mathematical expression that consists of variables and coefficients. for example x^2 - 4x + 7.In the Polynomial linked list, the coefficients and exponents of the polynomial are defined as the data node of the list.



Algorithm for polynomial addition:

- Include libraries and create structure of the node.
- Initialize three head pointers head1, head2, head3, for the two input polynomials and resultant polynomial. Then, we create two linked list by inserting coefficients and exponent values to both the linked lists.
- When polynomials are created in sorted order, next step is to add them according to the exponent value. Consider pointer temp1 and temp2 for traversing poly1 and poly2.Use temp3 pointer for the resultant polynomial. We will consider three cases:
    - Case 1: If exponent value of poly1 is equal to the exponent value of poly 2, then add their coefficient terms directly and output their addition. Move temp1, temp2 and temp3 to the next position.
    - Case 2: If exponent value of poly1 is greater than the exponent value of poly 2, then output poly1 as it is. Move temp1, temp3 to the next position.
    - Case 3: If exponent value of poly1 is less than the exponent value of poly 2, then output poly2 as it as. Move temp2, temp3 to the next position
    - Continue to append the remaining nodes(temp1->next==temp2->next==NULL) from or until we finish the calculation on all nodes

Display the nodes (Traversing): Displaying all the nodes of a stack needs traversing all the nodes of the linked list organized in the form of stack. For this purpose, we need to follow the following steps.

- Copy the head pointer into a temporary pointer.
- Move the temporary pointer through all the nodes of the list and print the value field attached to every node.

Code:

```
import java.util.*;

class Node

{

   int coeff;

   int power;

   Node next;


   Node(int coeff,int power)

   {

      this.coeff = coeff;

      this.power=power;

      this.next = null;

   }

}


public class LinkedList

{


   Node head=null;
```

```
  public void insertend(int coeff, int power) {
    Node newNode = new Node(coeff, power);
    if (head == null) {
        head = newNode;
    } else {
        Node temp = head;
        while (temp.next != null) {
            temp = temp.next;
        }
        temp.next = newNode;
    }
    System.out.println("Inserted: " + coeff + "x^" + power);
}


public void display() {
    if (head == null) {
        System.out.println("List is empty");
        return;
    }
    Node temp = head;
    System.out.print("Polynomial: ");
    while (temp != null) {
        System.out.print(temp.coeff + "x^" + temp.power + (temp.next != null ? " + " : ""));
        temp = temp.next;
```

```
    }
    System.out.println();
}

  public LinkedList add(LinkedList list1, LinkedList list2) {
    LinkedList result = new LinkedList();
    Node current1 = list1.head;
    Node current2 = list2.head;

    while (current1 != null || current2 != null) {
       if (current1 == null) {
          result.insertend(current2.coeff, current2.power);
          current2 = current2.next;
       } else if (current2 == null) {
          result.insertend(current1.coeff, current1.power);
          current1 = current1.next;
       } else if (current1.power == current2.power) {
          result.insertend(current1.coeff + current2.coeff, current1.power);
          current1 = current1.next;
          current2 = current2.next;
       } else if (current1.power > current2.power) {
          result.insertend(current1.coeff, current1.power);
          current1 = current1.next;
       } else {
          result.insertend(current2.coeff, current2.power);
```

```
            current2 = current2.next;
        }
    }
    return result;
}

public static void main(String[] args)
{
    System.out.println("Jay and Janay");
    LinkedList list1 = new LinkedList();
    LinkedList list2 = new LinkedList();
    LinkedList list3 = new LinkedList();

    Scanner sc = new Scanner(System.in);
    while (true)
    {
        System.out.println("1. Insert in 1 \n2. Insert in 2\n3. Display\n4. Addition\n5. Exit\n");
        int choice = sc.nextInt();
        switch (choice)
        {
            case 1:
                System.out.println("Enter value to insert:");
                int value1 = sc.nextInt();
                System.out.println("Enter power to insert:");
                int value2 = sc.nextInt();
```

```
                list1.insertend(value1,value2);
                break;
            case 2:
                System.out.println("Enter value to insert:");
                value1 = sc.nextInt();
                System.out.println("Enter power to insert:");
                value2 = sc.nextInt();
                list2.insertend(value1,value2);
                break;
            case 3:
                 System.out.println("First Polynomial :");
                 list1.display();
                 System.out.println("Second Polynomial :");
                 list2.display();
                 break;
            case 4:
                System.out.println("The addition of two polynomial is :");
                list3 = list1.add(list1,list2);
                list3.display();
            case 5:
                return;
        }
    }
  }
}
```

Department of Electronics & Telecommunication Engineering
Data Structures & Algorithms Lab ((DJS22EL505)

```
Jay and Janay
1. Insert in 1
2. Insert in 2
3. Display
4. Addition
5. Exit

1
Enter value to insert:
10
Enter power to insert:
3
Inserted: 10x^3
1. Insert in 1
2. Insert in 2
3. Display
4. Addition
5. Exit

1
Enter value to insert:
20
Enter power to insert:
2
Inserted: 20x^2
1. Insert in 1
2. Insert in 2
3. Display
4. Addition
5. Exit

1
Enter value to insert:
0
Enter power to insert:
1
Inserted: 0x^1
1. Insert in 1
2. Insert in 2
3. Display
4. Addition
5. Exit

3
```

```
First Polynomial :
Polynomial: 10x^3 + 20x^2 + 0x^1
Second Polynomial :
List is empty
1. Insert in 1
2. Insert in 2
3. Display
4. Addition
5. Exit

2
Enter value to insert:
5
Enter power to insert:
3
Inserted: 5x^3
1. Insert in 1
2. Insert in 2
3. Display
4. Addition
5. Exit

2
Enter value to insert:
20
Enter power to insert:
2
Inserted: 20x^2
1. Insert in 1
2. Insert in 2
3. Display
4. Addition
5. Exit

2
Enter value to insert:
30
Enter power to insert:
1
Inserted: 30x^1
1. Insert in 1
2. Insert in 2
3. Display
4. Addition
```

```
Inserted: 30x^1
1. Insert in 1
2. Insert in 2
3. Display
4. Addition
5. Exit

3
First Polynomial :
Polynomial: 10x^3 + 20x^2 + 0x^1
Second Polynomial :
Polynomial: 5x^3 + 20x^2 + 30x^1
1. Insert in 1
2. Insert in 2
3. Display
4. Addition
5. Exit

4
The addition of two polynomial is :
Inserted: 15x^3
Inserted: 40x^2
Inserted: 30x^1
Polynomial: 15x^3 + 40x^2 + 30x^1


...Program finished with exit code 0
Press ENTER to exit console.
```

Result and Conclusion: