OUTPUT

```
Enterpolynomial 1
Enter number of terms : 4.
Term 1 | Coff and Exp : 4 3
Term2 | Coff and Exp : 3 2
Term 3 | Coff and Exp : 12 1
Term 4 | Coff and Exp : 1 0
Enter Polynomial 2
Enter number of terms : 3
Term1 | Coff and Exp : 6 3
Term2 | Coff and Exp : 8 1
Term 3 | Coff and Exp : 9 0
Poly1
+4x^3 + 3x^2 + 12x^1 + 10x^0
Poly2
+6x^3 + 8x^1 + 9x^0
Polynomial Sum
+10x^3 + 3x^2 + 20x^1 + 10x^0
```

Q. Implement polynomial addition using a singly linked list

ALGORITHM

1. Declare structure Term, with coff, exp and pointer to next Term.

```
I. polyAdder(poly1, poly2)
    1. res = lastNode = NULL
    2. i = poly1 ; j = poly2
    3. while (i not NULL OR j not NULL):
        1. Create new Term, newNode
        2. if (exp at i == exp at j):
            1. newNode -> coff = i -> coff + j -> coff
            2. newNode -> exp = i -> exp
            3. i = i -> next
            4. j = j -> next
        3. else if (exp of i > exp of j OR j == NULL):
            1. newNode -> coff = i -> coff
            2. newNode -> exp = i -> exp
            3. i = i -> next
        4. else:
            1. newNode -> coff = j -> coff
            2. newNode -> exp = j -> exp
            3. j = j -> next
        5. if (lastNode == NULL):
            res = newNode
        6. else:
            lastNode -> next = newNode
        7. lastNode = newNode
        8. newNode -> next = NULL
    4. return res.
```

II. **polyRead()**

```
1. INPUT length as len
2. i = 0 ; res = NULL
3. while(i < len):
    1. Create new Term, newNode
    2. INPUT coff, exp into newNode
    3. if (prev==NULL):
           res = newNode
    4. else:
           prev->next = newNode
           prev = newNode
    5. newNode->next = NULL
    6. i = i+1
4. return res.
```

III. **polyShow(poly)**

```
1. p=poly
2. while(p≠NULL):
    1. DISPLAY coff, exp
    2. p = p->next
```

IV. **main**

```
1. Start
2. using polyRead, get poly1 and poly2
3. Call polyAddn using poly1 and poly2 as paramnetels,
   store return value in polysum.
4. DISPLAY poly1, poly2, polysum using polyShow.
5. End.
```

RESULT

Theprogram is executed succesfully and output is obtained.