

```

1: //Adding two polynomials using linkedlist
2: /*Given two polynomial numbers represented by a linkedlist.
3: Write a function that adds these lists means add the coeffi
4: which have same variable powers.*/
5:
6: #include<stdio.h>
7: #include<stdlib.h>
8:
9: struct Node
10: {
11:     int coeff;
12:     int pow;
13:     struct Node *next;
14: };
15:
16: void create_node(int x, int y, struct Node **temp)
17: {
18:     struct Node *r, *z;
19:     z = *temp;
20:     if(z == NULL)
21:     {
22:         r = (struct Node*)malloc(sizeof(struct Node));
23:         r->coeff = x;
24:         r->pow = y;
25:         *temp = r;
26:         r->next = (struct Node*)malloc(sizeof(struct Node));
27:         r = r->next;
28:         r->next = NULL;
29:     }
30:     else
31:     {
32:         r->coeff = x;
33:         r->pow = y;
34:         r->next = (struct Node*)malloc(sizeof(struct Node));
35:         r = r->next;
36:         r->next = NULL;
37:     }
38: }
39:

```

```

40: void polyadd(struct Node *poly1, struct Node *poly2, struct Node *
41: {
42: while(poly1->next && poly2->next)
43: {
44: if(poly1->pow > poly2->pow)
45: {
46: poly->pow = poly1->pow;
47: poly->coeff = poly1->coeff;
48: poly1 = poly1->next;
49: }
50: else if(poly1->pow < poly2->pow)
51: {
52: poly->pow = poly2->pow;
53: poly->coeff = poly2->coeff;
54: poly2 = poly2->next;
55: }
56: else
57: {
58: poly->pow = poly1->pow;
59: poly->coeff = poly1->coeff+poly2->coeff;
60: poly1 = poly1->next;
61: poly2 = poly2->next;
62: }
63: poly->next = (struct Node *)malloc(sizeof(struct Node));
64: poly = poly->next;
65: poly->next = NULL;
66: }
67: while(poly1->next || poly2->next)
68: {
69: if(poly1->next)
70: {
71: poly->pow = poly1->pow;
72: poly->coeff = poly1->coeff;
73: poly1 = poly1->next;
74: }
75: if(poly2->next)
76: {
77: poly->pow = poly2->pow;
78: poly->coeff = poly2->coeff;

```

```

79: poly2 = poly2->next;
80: }
81: poly->next = (struct Node *)malloc(sizeof(struct Node));
82: poly = poly->next;
83: poly->next = NULL;
84: }
85: }
86:
87: void show(struct Node *node)
88: {
89: while(node->next != NULL)
90: {
91: printf("%dx^d", node->coeff, node->pow);
92: node = node->next;
93: if(node->next != NULL)
94: printf(" + ");
95: }
96: }
97:
98: int main()
99: {
100: struct Node *poly1 = NULL, *poly2 = NULL, *poly = NULL;
101: // Create first list of 5x^2 + 4x^1 + 2x^0
102: create_node(5,2,&poly1);
103: create_node(4,1,&poly1);
104: create_node(2,0,&poly1);
105: // Create second list of 5x^1 + 5x^0
106: create_node(5,1,&poly2);
107: create_node(5,0,&poly2);
108: printf("1st Number: ");
109: show(poly1);
110: printf("\n2nd Number: ");
111: show(poly2);
112: poly = (struct Node *)malloc(sizeof(struct Node));
113: // Function add two polynomial numbers
114: polyadd(poly1, poly2, poly);
115: // Display resultant List
116: printf("\nAdded polynomial: ");
117: show(poly);

```

```
118: return 0;  
119: }  
120:
```