Input:

1st number = 5x^2 + 4x^1 + 2x^0

2nd number = 5x^1 + 5x^0

Output:

5x^2 + 9x^1 + 7x^0

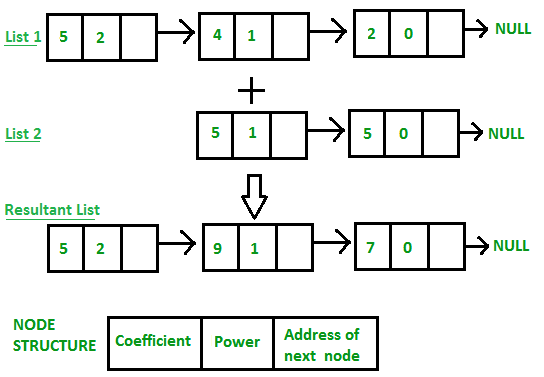
Input:

1st number = 5x^3 + 4x^2 + 2x^0

2nd number = 5x^1 + 5x^0

Output:

5x^3 + 4x^2 + 5x^1 + 7x^0

[](https://media.geeksforgeeks.org/wp-content/uploads/Addition-of-two-polynomial.png)

|  |
| --- |
| // program for addition of two polynomials  // using Linked Lists      // Node structure containing power and coefficient of variable  struct Node  {      int coeff;      int pow;      struct Node \*next;  };      // Function Adding two polynomial numbers  void polyadd(struct Node \*poly1, struct Node \*poly2, struct Node \*poly)  {  while(poly1->next && poly2->next)      {          // If power of 1st polynomial is greater then 2nd, then store 1st as it is          // and move its pointer          if(poly1->pow > poly2->pow)          {              poly->pow = poly1->pow;              poly->coeff = poly1->coeff;              poly1 = poly1->next;          }            // If power of 2nd polynomial is greater then 1st, then store 2nd as it is          // and move its pointer          else if(poly1->pow < poly2->pow)          {              poly->pow = poly2->pow;              poly->coeff = poly2->coeff;              poly2 = poly2->next;          }            // If power of both polynomial numbers is same then add their coefficients          else          {              poly->pow = poly1->pow;              poly->coeff = poly1->coeff+poly2->coeff;              poly1 = poly1->next;              poly2 = poly2->next;          }            // Dynamically create new node          poly->next = (struct Node \*)malloc(sizeof(struct Node));          poly = poly->next;          poly->next = NULL;      }  while(poly1->next || poly2->next)      {          if(poly1->next)          {              poly->pow = poly1->pow;              poly->coeff = poly1->coeff;              poly1 = poly1->next;          }          if(poly2->next)          {              poly->pow = poly2->pow;              poly->coeff = poly2->coeff;              poly2 = poly2->next;          }          poly->next = (struct Node \*)malloc(sizeof(struct Node));          poly = poly->next;          poly->next = NULL;      }  } |