// 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;

    }

}

// Display Linked list

void show(struct Node \*node)

{

while(node->next != NULL)

    {

    printf("%dx^%d", node->coeff, node->pow);

    node = node->next;

    if(node->next != NULL)

        printf(" + ");

    }

}