



Linkedlist based polynomial addition



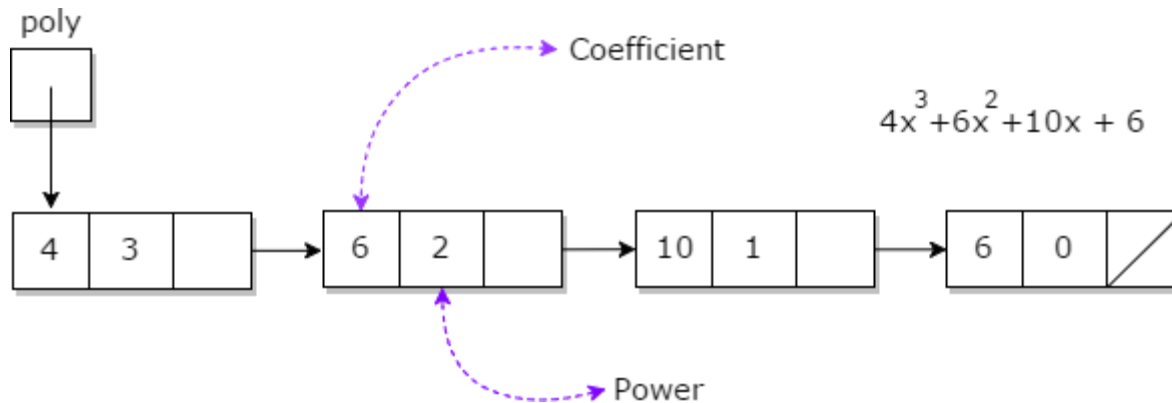
Polynomial addition

- A polynomial $p(x)$ is the expression in variable x which is in the form $(ax^n + bx^{n-1} + \dots + jx + k)$
- where a, b, c, \dots, k are real numbers
- 'n' is non negative integer, which is called the degree of polynomial.
- An essential characteristic of the polynomial is that each term in the polynomial expression consists of two parts:
 - ✓ one is the coefficient
 - ✓ other is the exponent



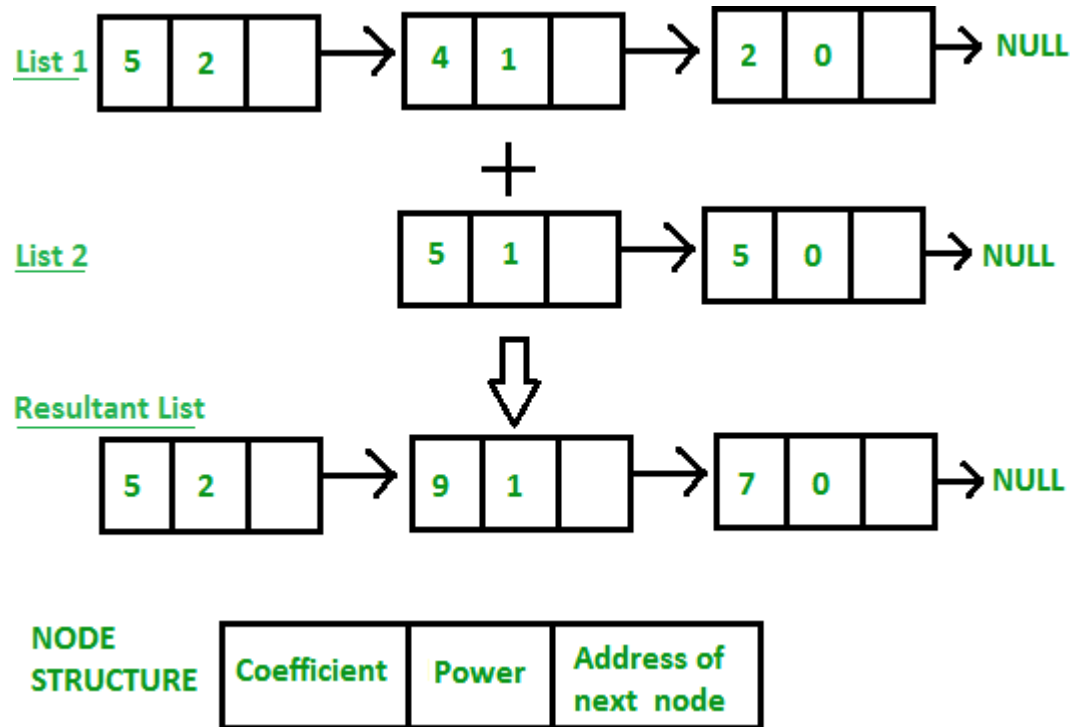
Example:

- $4x^3 + 6x^2 + 10x + 6$
- here 4,6 ,10 and 6 are coefficients
- 3,2 and 1 is its exponential value.





Addition of two polynomial





Algorithm to add two polynomials using linked list



- Let p and q be the two polynomials represented by linked lists

Step1: while p and q are not null, repeat step 2.

Step2: If powers of the two terms are equal then if the terms do not cancel then insert the sum of the terms into the sum Polynomial

- Advance p
- Advance q
- Else if the power of the first polynomial > power of second
- Then insert the term from first polynomial into sum polynomial
- Advance p
- Else insert the term from second polynomial into sum polynomial
- Advance q

Step3: copy the remaining terms from the non empty polynomial into the sum polynomial.



PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
```

```
typedef struct node
{
    int coef;
    int exp;
    struct node* next;
} node;
```

```
void get_input(node** head)
{
    node* temp,*ptr;
    ptr = *head;
    temp = (node*)malloc(sizeof(node));

    printf("\nEnter coef : ");
    scanf("%d",&(temp->coef));
    printf("\nEnter exp : ");
    scanf("%d",&(temp->exp));
```





```
if(NULL == *head)
{
    *head = temp;
    (*head)->next = NULL;
}
else
{
    while(NULL != ptr->next)
    {
        ptr = ptr->next;
    }
    ptr->next = temp;
    temp->next = NULL;
}
}
void display(node* head)
{
    while(head->next != NULL)
    {
        printf("(%d.x^%d)+",head->coef,head->exp);
        head = head->next;
    }
    printf("(%d.x^%d)",head->coef,head->exp);
    printf("\n");
}
```



```
void add(node* poly, node* poly1, node* poly2 ) //poly==result
{
    while(poly1->next && poly2->next)
    {
        if(poly1->exp > poly2->exp)
        {
            poly->coef = poly1->coef;
            poly->exp = poly1->exp;
            poly1 = poly1->next;
        }
        else if(poly1->exp < poly2->exp)
        {
            poly->coef = poly2->coef;
            poly->exp = poly2->exp;
            poly2 = poly2->next;
        }
        else
        {
            poly->coef = poly1->coef + poly2->coef;
            poly->exp = poly1->exp;
            poly1 = poly1->next;
            poly2 = poly2->next;
        }
        poly->next = (node*)malloc(sizeof(node));
        poly = poly->next;
        poly->next = NULL;
    }
}
```





```
int main()
{
    node* head1 = NULL;
    node* head2 = NULL;
    node* head = (node*)malloc(sizeof(node));
    int ch;
    do
    {
        get_input(&head1);
        printf("\nEnter more node in poly1? (1,0) :");
        scanf("%d",&ch);
    }while(ch);
    do
    {
        get_input(&head2);
        printf("\nEnter more node in poly2? (1,0) :");
        scanf("%d",&ch);
    }while(ch);
    add(head,head1,head2);
    display(head1);
    display(head2);
    display(head);
    return 0;
}
```