

Lecture-08

Polynomial List

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 fall in the category of real numbers and ' n ' is non negative integer, which is called the degree of polynomial.

An important characteristics of polynomial is that each term in the polynomial expression consists of two parts:

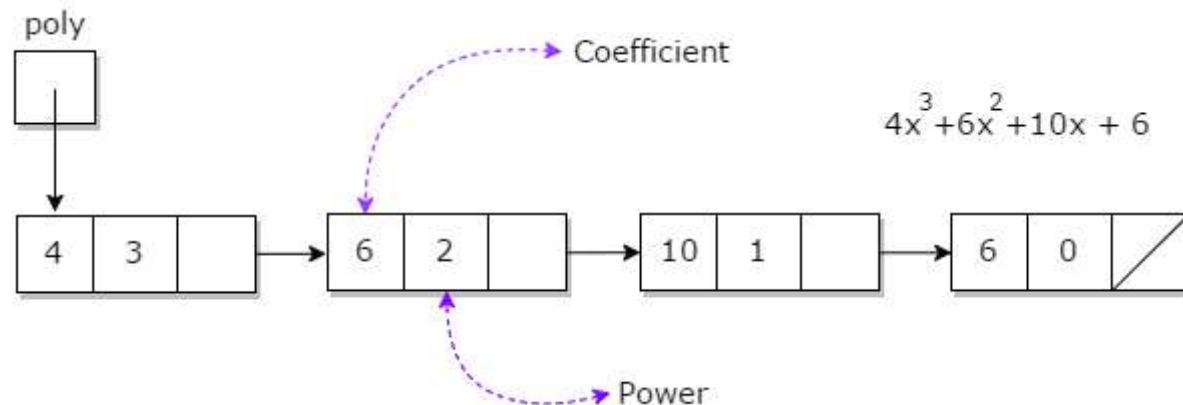
- one is the coefficient
- other is the exponent

Example:

$10x^2 + 26x$, here 10 and 26 are coefficients and 2, 1 are its exponential value.

Points to keep in Mind while working with Polynomials:

- The sign of each coefficient and exponent is stored within the coefficient and the exponent itself
- Additional terms having equal exponent is possible one
- The storage allocation for each term in the polynomial must be done in ascending and descending order of their exponent



Representation of Polynomial

Polynomial can be represented in the various ways. These are:

- By the use of arrays
- By the use of Linked List

Representation of Polynomials using Arrays

There may arise some situation where you need to evaluate many polynomial expressions and perform basic arithmetic operations like: addition and subtraction with those numbers. For this you will have to get a way to represent those polynomials. The simple way is to represent a polynomial with degree ' n ' and store the coefficient of $n+1$ terms of the polynomial in array. So every array element will consists of two values:

- Coefficient and
- Exponent

Representation of Polynomial Using Linked Lists

A polynomial can be thought of as an ordered list of non zero terms. Each non zero term is a two tuple which holds two pieces of information:

- The exponent part
- The coefficient part

Adding two polynomials using Linked List

Given two polynomial numbers represented by a linked list. Write a function that add these lists means add the coefficients who have same variable powers.

Example:

Input:

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

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

Output:

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

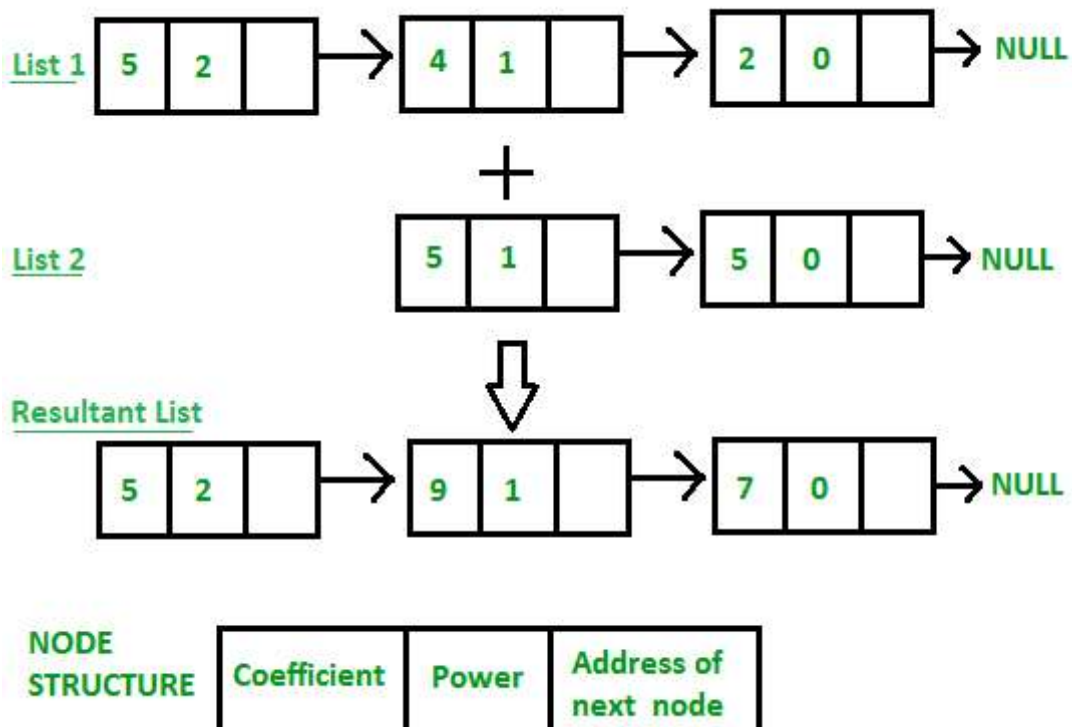
Input:

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

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

Output:

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



```

struct Node
{
    int coeff;

    int pow;

    struct Node *next;
};

void create_node(int x, int y, struct Node **temp)
{
    struct Node *r, *z;

    z = *temp;

    if(z == NULL)
    {
        r = (struct Node*)malloc(sizeof(struct Node));

        r->coeff = x;

        r->pow = y;

        *temp = r;

        r->next = (struct Node*)malloc(sizeof(struct Node));

        r = r->next;

        r->next = NULL;
    }

    else
    {
        r->coeff = x;

        r->pow = y;
    }
}

```

```

    r->next = (struct Node*)malloc(sizeof(struct Node));

    r = r->next;

    r->next = NULL;
}

}

void polyadd(struct Node *poly1, struct Node *poly2, struct Node *poly)
{
    while(poly1->next && poly2->next)
    {
        if(poly1->pow > poly2->pow)
        {
            poly->pow = poly1->pow;
            poly->coeff = poly1->coeff;
            poly1 = poly1->next;
        }
        else if(poly1->pow < poly2->pow)
        {
            poly->pow = poly2->pow;
            poly->coeff = poly2->coeff;
            poly2 = poly2->next;
        }
        else
        {
            poly->pow = poly1->pow;
            poly->coeff = poly1->coeff+poly2->coeff;

```

```

    poly1 = poly1->next;
    poly2 = poly2->next;
}
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;
}

```

```

}

void show(struct Node *node)
{
while(node->next != NULL)
    {
        printf("%dx^%d", node->coeff, node->pow);
        node = node->next;
        if(node->next != NULL)
            printf(" + ");
    }
}

int main()
{
    struct Node *poly1 = NULL, *poly2 = NULL, *poly = NULL;
    // Create first list of  $5x^2 + 4x^1 + 2x^0$ 
    create_node(5,2,&poly1);
    create_node(4,1,&poly1);
    create_node(2,0,&poly1);
    // Create second list of  $5x^1 + 5x^0$ 
    create_node(5,1,&poly2);
    create_node(5,0,&poly2);
    printf("1st Number: ");
    show(poly1);
    printf("\n2nd Number: ");
    show(poly2);
}

```

```
poly = (struct Node *)malloc(sizeof(struct Node));  
// Function add two polynomial numbers  
polyadd(poly1, poly2, poly);  
// Display resultant List  
printf("\nAdded polynomial: ");  
show(poly);  
return 0;  
}
```

Output:

1st Number: $5x^2 + 4x^1 + 2x^0$

2nd Number: $5x^1 + 5x^0$

Added polynomial: $5x^2 + 9x^1 + 7x^0$