

Consider the following polynomials

$$4x^3 + 3x^2 + 1 \text{ ————— } \textcircled{1}$$

$$5x^3 + 7x + 5 \text{ ————— } \textcircled{2}$$

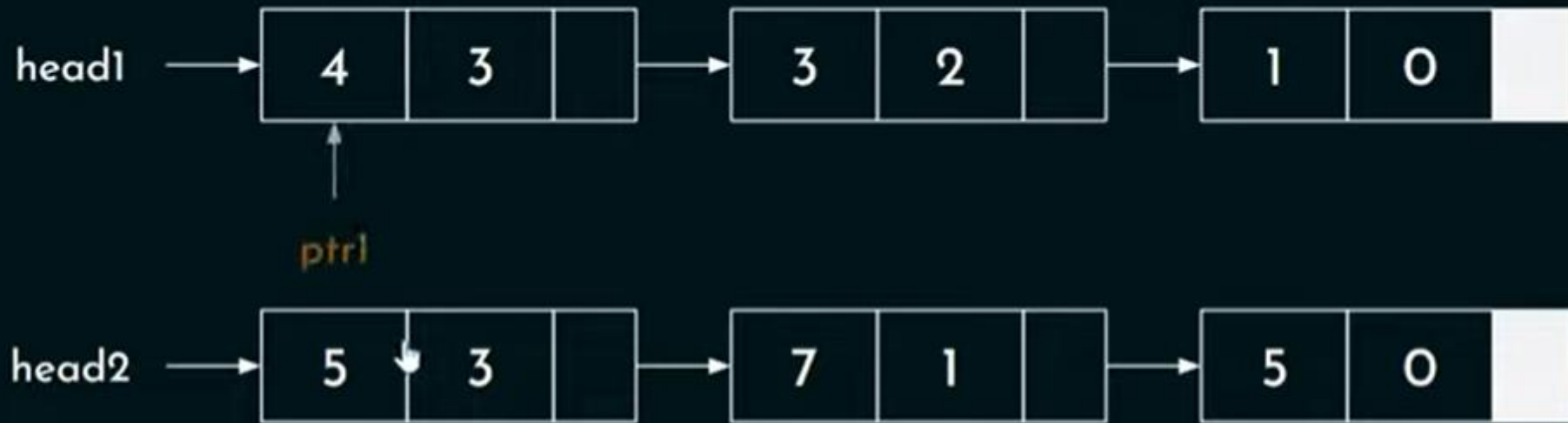
Each term of the polynomial $\textcircled{1}$ must be multiplied with each term of the polynomial $\textcircled{2}$

Multiplying each term means multiplying their coefficients and adding their exponents.

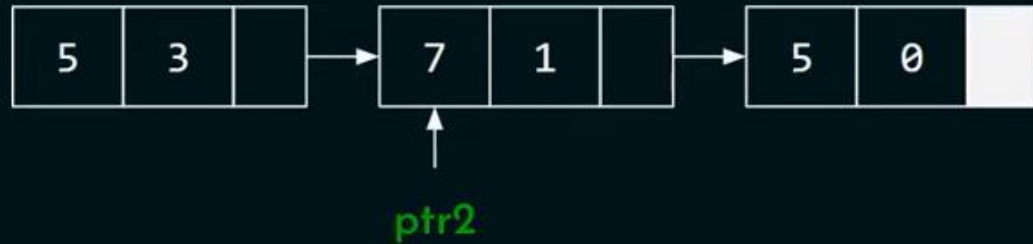
$$(4 \times 5)x^{3+3} + (4 \times 7)x^{3+1} + (4 \times 5)x^{3+0} + (3 \times 5)x^{2+3} + (3 \times 7)x^{2+1} + (3 \times 5)x^{2+0} + (1 \times 5)x^{0+3} \\ + (1 \times 7)x^{0+1} + (1 \times 5)x^{0+0}$$

$$20x^6 + 28x^4 + 20x^3 + 15x^5 + 21x^3 + 15x^2 + 5x^3 + 7x + 5$$

Resultant Polynomial



We need two pointers (**ptr1** and **ptr2**) for traversal.
We also need a nested loop as each term of the first polynomial must be multiplied with every term of the second polynomial.



```
int res1, res2;
struct node* head3 = NULL;
while(ptr1 != NULL)
{
    while(ptr2 != NULL)
    {
        res1 = ptr1->coeff * ptr2->coeff;
        res2 = ptr1->expo + ptr2->expo;
        head3 = insert(head3, res1, res2);
        ptr2 = ptr2->link;
    }
}
```

28	4
res1	res2



head1

ptr1



ptr2

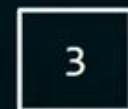


head3

```
int res1, res2;
struct node* head3 = NULL;
while(ptr1 != NULL)
{
    ptr2 = head2;
    while(ptr2 != NULL)
    {
        res1 = ptr1->coeff * ptr2->coeff;
        res2 = ptr1->expo + ptr2->expo;
        head3 =insert(head3, res1, res2);
        ptr2 = ptr2->link;
    }
    ptr1 = ptr1->link;
}
```



res1



res2

$$4x^3 + 3x^2 + 1 \text{ ——— } \textcircled{1}$$

$$5x^3 + 7x + 5 \text{ ——— } \textcircled{2}$$

$$(4 \times 5)x^{3+3} + (4 \times 7)x^{3+1} + (4 \times 5)x^{3+0} + (3 \times 5)x^{2+3} + (3 \times 7)x^{2+1} + (3 \times 5)x^{2+0} + (1 \times 5)x^{0+3} \\ + (1 \times 7)x^{0+1} + (1 \times 5)x^{0+0}$$

$$20x^6 + 28x^4 + 20x^3 + 15x^5 + 21x^3 + 15x^2 + 5x^3 + 7x + 5$$

$$20x^6 + 15x^5 + 28x^4 + 20x^3 + 21x^3 + 5x^3 + 15x^2 + 7x + 5$$

We will get this polynomial
after executing the code
because of **insert function**.



Other Way


```

NODE* POLY_MULT(NODE *A, NODE *B)
{

```

```

    NODE *p, *q, *PROD, *SOP;
    int C, E;
    PROD=NULL;
    SOP=NULL;

```

```

    p=A;
    while(p!=NULL)
    {

```

```

        q=B;
        while(q!=NULL)
        {

```

```

            C = p->COEF * q->COEF;
            E = p->EXP + q->EXP;
            InsLast(&PROD,C,E);
            q = q->NEXT;
        }

```

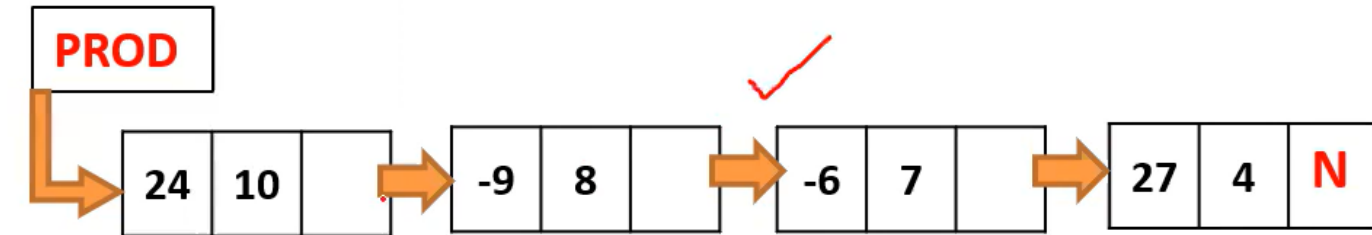
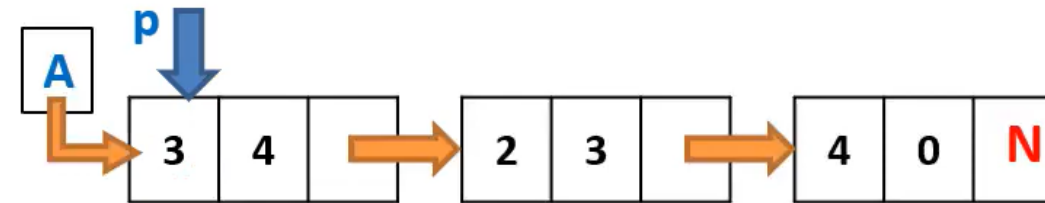
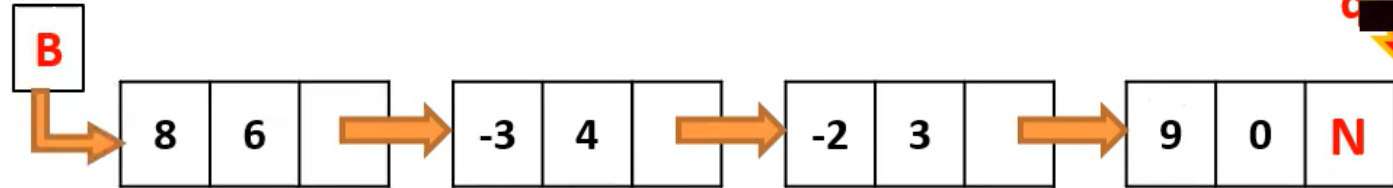
```

        SOP = PADD(SOP,PROD);
    }
}

```

$$B = 8X^6 - 3X^4 - 2X^3 + 9$$

$$A = 3X^4 + 2X^3 + 4$$



```
NODE* POLY_MULT(NODE *A, NODE *B)
```

```
{
```

```
    NODE *p, *q, *PROD, *SOP;
```

```
    int C, E;
```

```
    PROD=NULL;
```

```
    SOP=NULL;
```

```
    p=A;
```

```
    while(p!=NULL)
```

```
    {
```

```
        q=B;
```

```
        while(q!=NULL)
```

```
        {
```

```
            C = p->COEF * q->COEF;
```

```
            E = p->EXP + q->EXP;
```

```
            InsLast(&PROD,C,E);
```

```
            q = q->NEXT;
```

```
        }
```

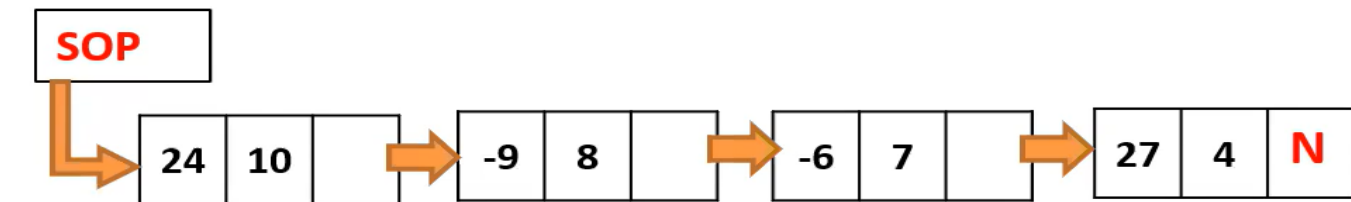
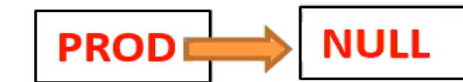
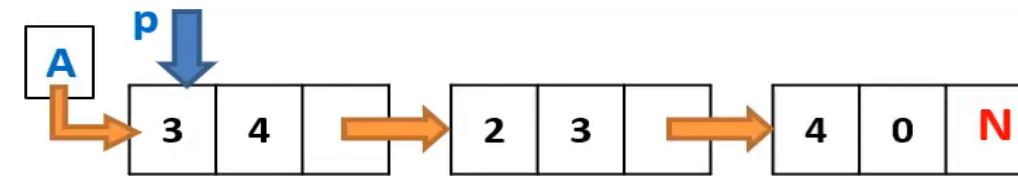
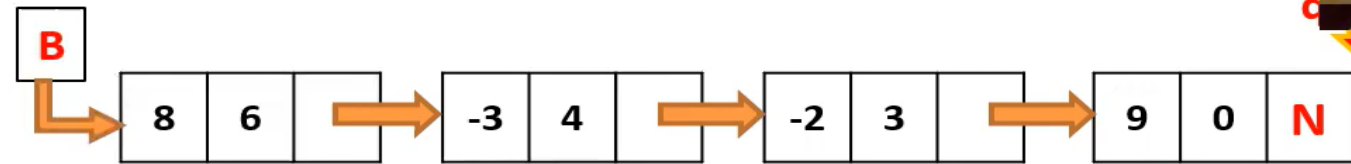
```
        SOP = PADD(SOP,PROD);
```

```
✓ delete_entire(&PROD);
```

```
        p = p->NEXT;
```

$$B = 8X^6 - 3X^4 - 2X^3 + 9$$

$$A = 3X^4 + 2X^3 + 4$$




```

NODE* POLY_MULT(NODE *A, NODE *B)

```

```

{

```

```

    NODE *p, *q, *PROD, *SOP;

```

```

    int C, E;

```

```

    PROD=NULL;

```

```

    SOP=NULL;

```

```

    p=A;

```

```

    while(p!=NULL)

```

```

    {

```

```

        q=B;

```

```

        while(q!=NULL)

```

```

        {

```

```

            C = p->COEF * q->COEF;

```

```

            E = p->EXP + q->EXP;

```

```

            InsLast(&PROD,C,E);

```

```

            q = q->NEXT;

```

```

        }

```

```

    }

```

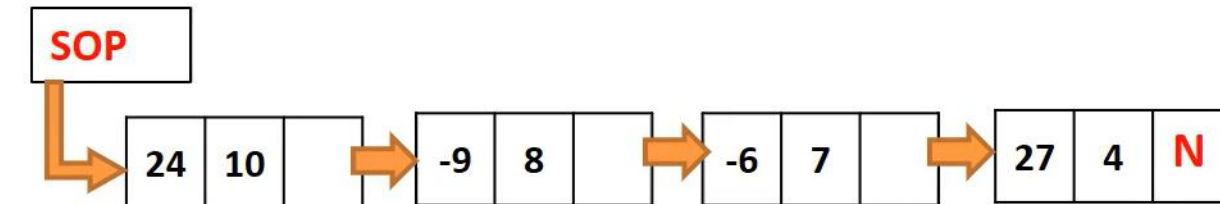
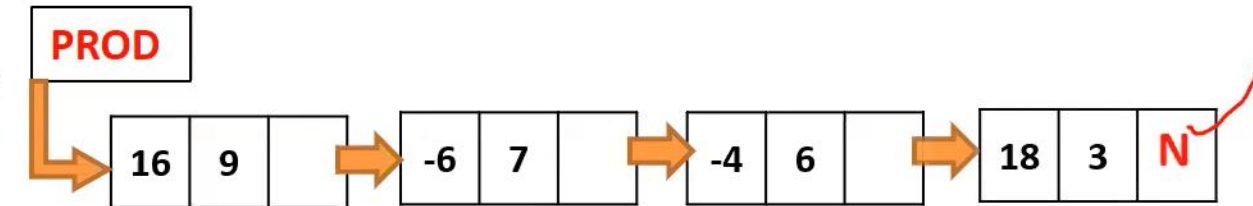
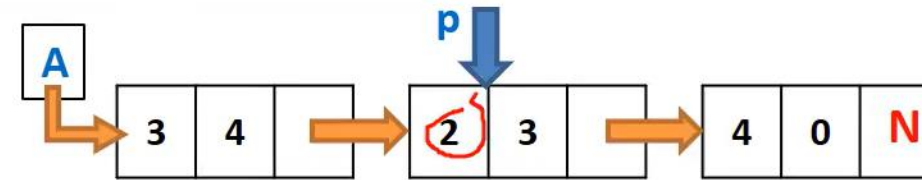
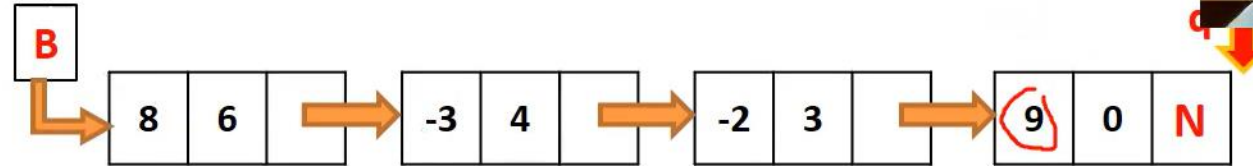
```

}

```

$$B = 8X^6 - 3X^4 - 2X^3 + 9$$

$$A = 3X^4 + 2X^3 + 4$$



```

NODE* POLY_MULT(NODE *A, NODE *B)
{

```

```

    NODE *p, *q, *PROD, *SOP;
    int C, E;
    PROD=NULL;
    SOP=NULL;

```

```

    p=A;
    while(p!=NULL)
    {

```

```

        q=B;
        while(q!=NULL)
        {

```

```

            C = p->COEF * q->COEF;
            E = p->EXP + q->EXP;
            InsLast(&PROD,C,E);
            q = q->NEXT;
        }

```

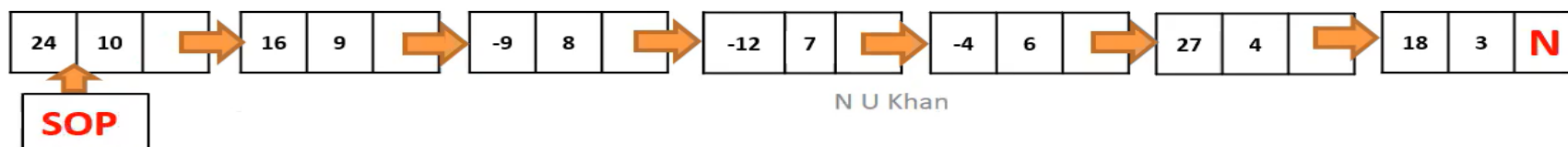
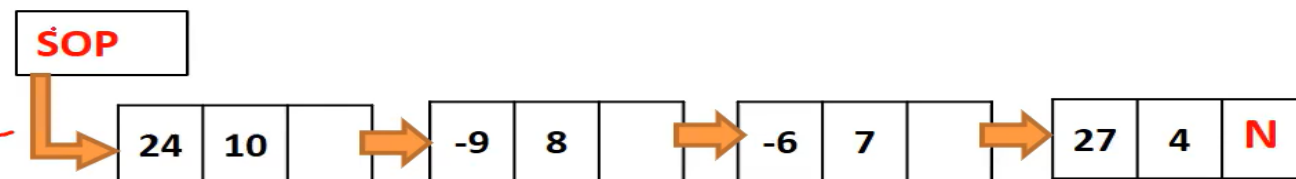
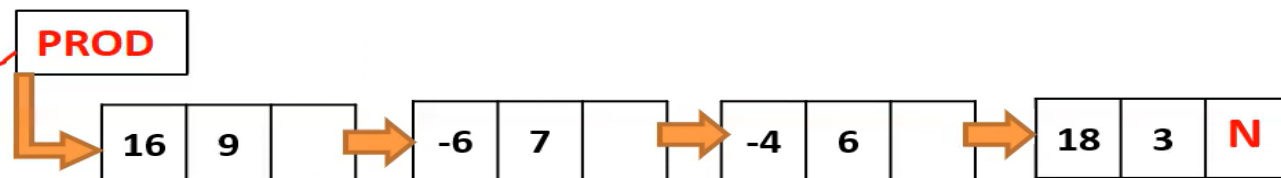
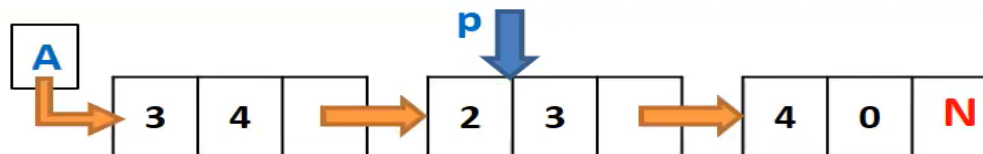
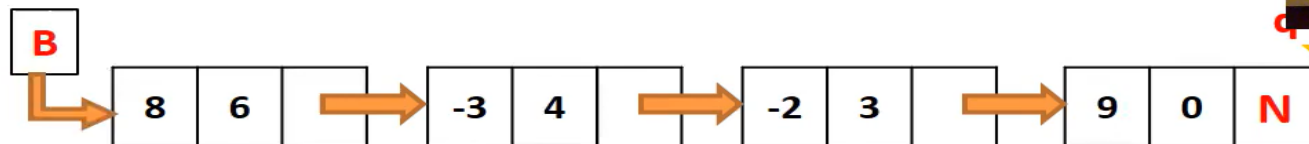
```

        SOP = PADD(SOP,PROD);
    }
}

```

$$B = 8X^6 - 3X^4 - 2X^3 + 9$$

$$A = 3X^4 + 2X^3 + 4$$



```

NODE* POLY_MULT(NODE *A, NODE *B)

```

```

{
    NODE *p, *q, *PROD, *SOP;
    int C, E;
    PROD=NULL;
    SOP=NULL;

```

```

    p=A;
    while(p!=NULL)
    {

```

```

        q=B;
        while(q!=NULL)
        {

```

```

            C = p->COEF * q->COEF;
            E = p->EXP + q->EXP;
            InsLast(&PROD,C,E);
            q = q->NEXT;
        }

```

```

        SOP = PADD(SOP,PROD);
        delete_entire(&PROD);
        p = p->NEXT;
    }

```

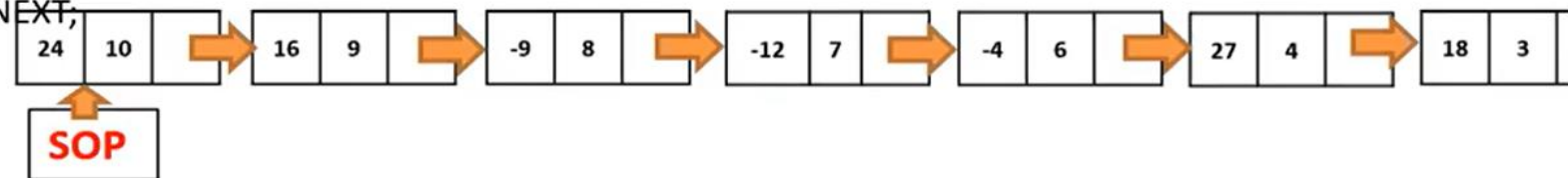
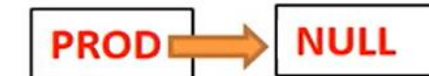
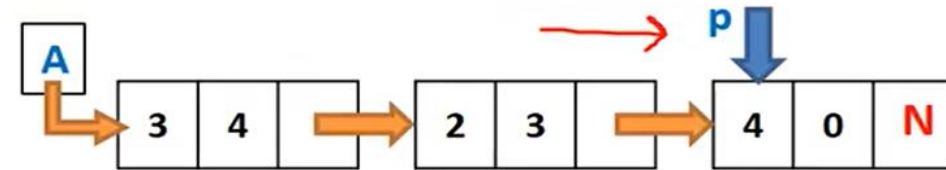
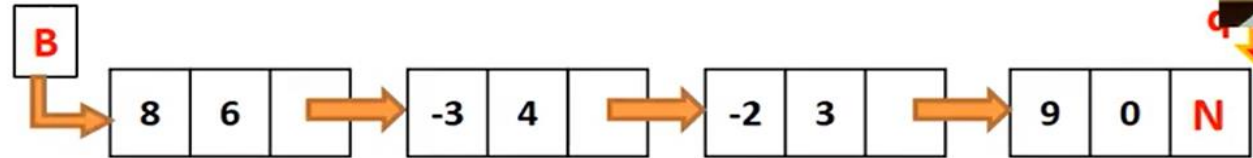
```

}

```

$$B = 8X^6 - 3X^4 - 2X^3 + 9$$

$$A = 3X^4 + 2X^3 + 4$$



```

NODE* POLY_MULT(NODE *A, NODE *B)
{

```

```

    NODE *p, *q, *PROD, *SOP;
    int C, E;
    PROD=NULL;
    SOP=NULL;

```

```

    p=A;
    while(p!=NULL)
    {

```

```

        q=B;
        while(q!=NULL)
        {

```

```

            C = p->COEF * q->COEF;
            E = p->EXP + q->EXP;
            InsLast(&PROD,C,E);
            q = q->NEXT;
        }

```

```

        SOP = PADD(SOP,PROD);
        delete_entire(&PROD);
        p = p->NEXT;
    }

```

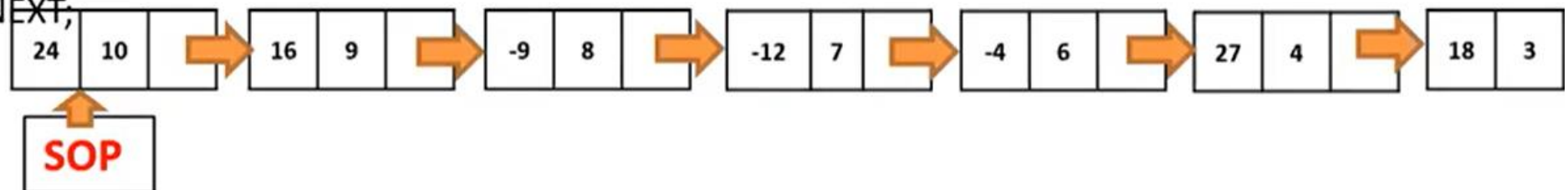
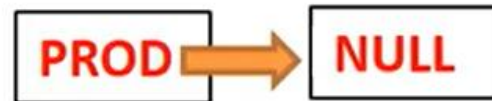
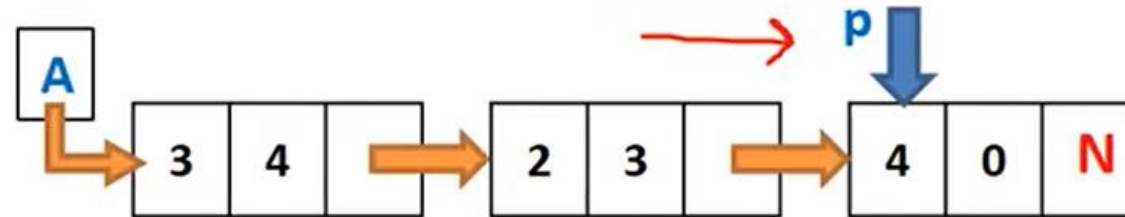
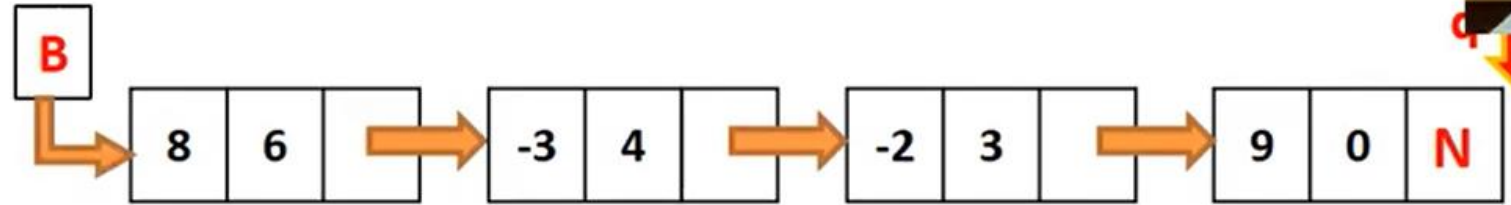
```

}

```

$$B = 8X^6 - 3X^4 - 2X^3 + 9$$

$$A = 3X^4 + 2X^3 + 4$$



NODE* POLY_MULT(NODE *A, NODE *B)

```
{
    NODE *p, *q, *PROD, *SOP;
    int C, E;
    PROD=NULL;
    SOP=NULL;
```

```
    p=A;
    while(p!=NULL)
```

```
    {
        q=B;
        while(q!=NULL)
```

```
        {
            C = p->COEF * q->COEF;
            E = p->EXP + q->EXP;
            InsLast(&PROD,C,E);
            q = q->NEXT;
```

```
        }
        SOP = PADD(SOP,PROD);
        delete_entire(&PROD);
        p = p->NEXT;
```

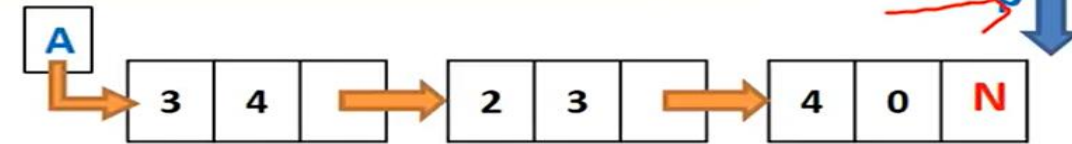
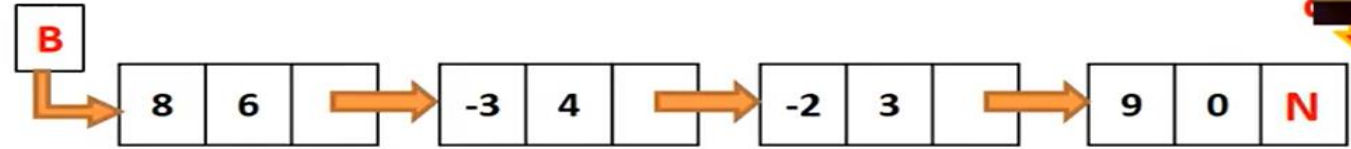
```
    }
    return(SOP);
```

SOP



$$B = 8X^6 - 3X^4 - 2X^3 + 9$$

$$A = 3X^4 + 2X^3 + 4$$




```
NODE* POLY_MULT(NODE *A, NODE *B )
```

```
{
```

```
    NODE *p, *q, *PROD, *SOP;
```

```
    int C, E;
```

```
    PROD=NULL;
```

```
    SOP=NULL;
```

```
    p=A;
```

```
    while(p!=NULL)
```

```
    {
```

```
        q=B;
```

```
        while(q!=NULL)
```

```
        {
```

```
            C=loc1->COEF * loc2->COEF;
```

```
            E=loc1->EXP + loc2->EXP;
```

```
            InsLast(&PROD,C,E);
```

```
            q = q->NEXT;
```

```
        }
```

```
        SOP = PADD(SOP,PROD);
```

```
        delete_entire(&PROD);
```

```
        p = p->NEXT;
```

```
    }
```

```
    return(SOP);
```

```
}
```

```
// Take first element of first polynomial
```

```
// Take first element of second polynomial
```

```
// Take entire 2nd polynomial
```

```
// Multiply their coefficients
```

```
// Add their exponent
```

```
// Create new polynomial with COEF is 'C' & EXP is 'E'
```

```
// Now take another element from a second polynomial
```

```
// add this new polynomial PROD to previous one and store to SOP
```

```
// Now delete entire newly created polynomial 'PROD'
```

```
// Now take 2nd element from a first polynomial
```

```
// return the multiplication to main function
```