```c
//Addtion & Multiplication of polynomials using stack
#include <stdio.h>
#include <stdlib.h>

struct term
{
    int coeff;
    int xpower;
    struct term *next;
};

struct polynomial_head
{
    struct term *start;
    int num_terms;
};

typedef struct term poly_term;
typedef struct polynomial_head poly_head;

poly_term *get_new_term()
{
    poly_term *t;
    t = (poly_term *)malloc(sizeof(poly_term));
    return t;
}

void term_insert(poly_head *h, int coeff, int xpower)
{
    poly_term *p,*q;
    p = get_new_term();
    p->coeff = coeff;
    p->xpower = xpower;
    if(h->start==NULL || (h->start != NULL && h->start->xpower < xpower))
    {
        p->next = h->start;
        h->start = p;
        h->num_terms++;
    }
    else if(h->start != NULL && h->start->xpower == xpower)
    {
        h->start->coeff += coeff;
    }
    else
    {
        q = h->start;
        while(q->next!=NULL && q->next->xpower > xpower)
            q = q->next;
        if(q->next!=NULL && q->next->xpower == xpower)
        {
            q->next->coeff += coeff;
        }
        else
        {
            p->next = q->next;
            q->next = p;
            h->num_terms++;
        }

    }

}

void displayPolynomial(poly_head h)
{
    poly_term *p = h.start;
    while(p!=NULL)
    {
        printf("(%d)x^%d", p->coeff, p->xpower);
```

```
        if(p->next!=NULL)
            printf(" + ");
        p = p->next;
    }
    printf("\n");
}

void new_poly(poly_head *h)
{
    int num_terms;
    printf("\nTerms in Polynomial : ");
    scanf("%d", &num_terms);
    printf("Enter The Polynomial : \n");
    int coeff, xpower;
    for (int i = 0; i < num_terms; ++i)
    {
        printf("Term %d : ", i+1);
        scanf("%d %d", &coeff, &xpower);
        term_insert(h, coeff, xpower);
    }
}

poly_head add(poly_head p1, poly_head p2)
{
    poly_head p3;
    p3.start = NULL;
    p3.num_terms = 0;
    poly_term *a = p1.start, *b = p2.start;
    while(a!=NULL && b!=NULL)
    {
        if(a->xpower > b->xpower)
        {
            term_insert(&p3, a->coeff, a->xpower);
            a = a->next;
        }
        else if(a->xpower < b->xpower)
        {
            term_insert(&p3, b->coeff, b->xpower);
            b = b->next;
        }
        else
        {
            term_insert(&p3, a->coeff+b->coeff, a->xpower);
            a = a->next;
            b = b->next;
        }
    }
    if(a!=NULL)
    {
        while(a!=NULL)
        {
            term_insert(&p3, a->coeff, a->xpower);
            a = a->next;
        }
    }
    else if(b!=NULL)
    {
        while(b!=NULL)
        {
            term_insert(&p3, b->coeff, b->xpower);
            b = b->next;
        }
    }
    return p3;
}

poly_head multiply(poly_head p1, poly_head p2)
{
    poly_head p3;
```

```
    p3.start = NULL;
    p3.num_terms = 0;

    poly_term *a=p1.start, *b;
    while(a!=NULL)
    {
        b = p2.start;
        while(b!=NULL)
        {
            term_insert(&p3, a->coeff*b->coeff, a->xpower+b->xpower);
            b = b->next;
        }
        a = a->next;
    }
    return p3;
}

int main()
{
    poly_head p1;
    p1.start = NULL;
    p1.num_terms = 0;
    new_poly(&p1);

    poly_head p2;
    p2.start = NULL;
    p2.num_terms = 0;
    new_poly(&p2);

    poly_head p3 = add(p1,p2);
    printf("P1 + P2 : \n");
    displayPolynomial(p3);
    printf("\n\n");
    poly_head p4 = multiply(p1,p2);
    printf("P1 * P2 : \n");
    displayPolynomial(p4);
}
/*
OUTPUT
Terms in Polynomial : 3
Enter The Polynomial :
Term 1 : 4 2
Term 2 : 5 1
Term 3 : 7 0

Terms in Polynomial : 2
Enter The Polynomial :
Term 1 : 1 2
Term 2 : 3 0
P1 + P2 :
(5)x^2 + (5)x^1 + (10)x^0


P1 * P2 :
(4)x^4 + (5)x^3 + (19)x^2 + (15)x^1 + (21)x^0
*/
```