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

typedef struct PolyNode* Polynomial;

struct PolyNode{
    int coef;
    int expon;
    Polynomial next;
};

Polynomial ReadPoly();
Polynomial Mult(Polynomial P1,Polynomial P2);
void PrintPoly(Polynomial PP);
Polynomial Add(Polynomial P1,Polynomial P2);

int main()
{
    Polynomial P1, P2, PP, PS;
    P1 = ReadPoly();
    P2 = ReadPoly();

    PP = Mult(P1, P2);
    PrintPoly(PP);

    PS = Add(P1, P2);
    PrintPoly(PS);

    return 0;
}


void Attach(int c, int e, Polynomial *pRear)
{
    Polynomial P;
    P = (Polynomial)malloc(sizeof(struct PolyNode));
    P->coef = c;
    P->expon = e;
    P->next = NULL;
    (*pRear)->next = P;
    *pRear = P;
}

Polynomial ReadPoly()
{
    Polynomial P, Rear, t;
    int c, e, N;
    scanf("%d", &N);
    P = (Polynomial)malloc(sizeof(struct PolyNode));
    P->next = NULL;
    Rear = P;
    while (N--)
    {
        scanf("%d %d", &c, &e);
        if (c != 0)
            Attach(c, e, &Rear);
    }
    t = P;
    P = P->next;
    free(t);
    return P;
}

Polynomial Add(Polynomial P1, Polynomial P2)
{
    Polynomial t1, t2;
    t1 = P1;
    t2 = P2;

    Polynomial P,t;
    P = (Polynomial)malloc(sizeof(struct PolyNode));
    P->next = NULL;
    Polynomial Rear;
    Rear = P;
    while (t1&&t2)
    {
        if (t1->expon==t2->expon)
        {
            if (t1->coef+t2->coef)
            {
                Attach(t1->coef + t2->coef, t1->expon, &Rear);
            }
            t1 = t1->next;
            t2 = t2->next;
        }
        else if (t1->expon>t2->expon)
        {
            Attach(t1->coef, t1->expon, &Rear);
            t1=t1->next;
        }else
        {
            Attach(t2->coef, t2->expon, &Rear);
            t2 = t2->next;
        }
    }
    while (t1)
    {
        Attach(t1->coef, t1->expon, &Rear);
        t1 = t1->next;
    }
    while (t2)
    {
        Attach(t2->coef, t2->expon, &Rear);
        t2 = t2->next;
    }
    t = P;
    P = P->next;
    free(t);
    return P;
}



Polynomial Mult(Polynomial P1, Polynomial P2)
{
    Polynomial P, Rear;
    Polynomial t1, t2, t;
    if (!P1||!P2)
    {
        return NULL;
    }
    t1 = P1;
    t2 = P2;
    P = (Polynomial)malloc(sizeof(struct PolyNode));
    Rear = P;
    while (t2)
    {

        Attach(t1->coef*t2->coef, t1->expon + t2->expon, &Rear);
        t2 = t2->next;
    }
    t1 = t1->next;
    while (t1)
    {
        t2 = P2;
        Rear = P;
        while (t2)
        {
            int e = t1->expon + t2->expon;
            int c = t1->coef * t2->coef;
            while (Rear->next&&Rear->next->expon>e)
            {
                Rear = Rear->next;
            }
            if (Rear->next&&Rear->next->expon==e)
            {
                if (Rear->next->coef+c)
                {
                    Rear->next->coef += c;
                }
                else
                {
                    t = Rear->next;
                    Rear->next = t->next;
                    free(t);
                }
            }
            else
            {
                t = (Polynomial)malloc(sizeof(struct PolyNode));
                t->coef = c;
                t->expon = e;
                t->next = Rear->next;
                Rear->next = t;

                Rear = Rear->next;
            }
            t2 = t2->next;
        }
        t1 = t1->next;
    }
    t2 = P;
    P = P->next;
    free(t2);

    return P;
}

void PrintPoly(Polynomial P)
{
    int flag = 0;
    if (!P)
    {
        printf("0 0\n");
        return;
    }
    while (P)
    {
        if (!flag)
        {
            flag = 1;
        }
        else
        {
            printf(" ");
        }
        printf("%d %d", P->coef, P->expon);
        P = P->next;
    }
    printf("\n");
}
```