

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

```
typedef struct Node{  
    int coef,expon;  
    struct Node *next;  
}Node,*LinkList;
```

```
int attach(int coef,int exponen,Node** lnode,Node *node){  
    node=(Node*)malloc(sizeof(Node));  
    node->coef=coef;  
    node->expon=exponen;  
    node->next=NULL;  
    (*lnode)->next=node;  
    (*lnode)=(*lnode)->next;  
}
```

```
LinkList init(){  
    LinkList list;  
    list=(LinkList)malloc(sizeof(Node));  
    list->next=NULL;  
    Node *lnode,*node;  
    lnode=list;  
    int num;  
    scanf("%d",&num);  
    for(int i=0;i<num;i++){  
        int coef,expon;  
        scanf("%d %d",&coef,&expon);  
        attach(coef,expon,&lnode,node);  
    }  
    return list;  
}
```

```
LinkList mult(LinkList list1,LinkList list2){  
    if(!list1->next||!list2->next){  
        return NULL;  
    }  
    LinkList list3;  
    list3=(LinkList)malloc(sizeof(Node));  
    list3->next=NULL;  
    Node *lnode1,*lnode2,*lnode3,*node;  
    lnode1=list1;  
    lnode2=list2;  
    lnode3=list3;  
    while(lnode2->next){
```

```

        attach(lnode1->next->coef*lnode2->next->coef,lnode1->next-
>expon+lnode2->next->expon,&lnode3,node);
        lnode2=lnode2->next;

    }
    lnode3=list3;
    lnode1=list1->next->next;
    lnode2=lnode2->next;
    while(lnode1){
        lnode2=list2->next;
        lnode3=list3;
        while(lnode2){
            int coef=lnode1->coef*lnode2->coef;
            int expon=lnode1->expon+lnode2->expon;

            while(lnode3->next&&lnode3->next->expon>expon){
                lnode3=lnode3->next;
            }

            if(lnode3->next&&lnode3->next->expon==expon){
                if(lnode3->next->coef+coef==0){
                    lnode3->next=lnode3->next->next;
                }else{
                    lnode3->next->coef=lnode3->next->coef+coef;
                }
            }else{
                node=(Node*)malloc(sizeof(Node));
                node->next=NULL;
                node->expon=expon;
                node->coef=coef;
                node->next=lnode3->next;
                lnode3->next=node;
            }
            lnode2=lnode2->next;
        }
        lnode1=lnode1->next;
    }
    return list3;

}

LinkedList add(LinkedList list1,LinkedList list2){
    if(!list1->next&&!list2->next){
        return NULL;
    }
    LinkedList list3;

```

```

list3=(LinkedList)malloc(sizeof(Node));
list3->next=NULL;
Node *node,*lnode1,*lnode2,*lnode3;
lnode1=list1->next;
lnode2=list2->next;
lnode3=list3;
while(lnode1&&lnode2){
    if(lnode1->expon<lnode2->expon){
        node=(Node*)malloc(sizeof(Node));
        node->coef=lnode2->coef;
        node->expon=lnode2->expon;
        node->next=NULL;
        lnode3->next=node;
        lnode3=lnode3->next;
        lnode2=lnode2->next;
    }

    if(lnode1->expon>lnode2->expon){
        node=(Node*)malloc(sizeof(Node));
        node->coef=lnode1->coef;
        node->expon=lnode1->expon;
        node->next=NULL;
        lnode3->next=node;
        lnode3=lnode3->next;
        lnode1=lnode1->next;
    }

    if(lnode1->expon==lnode2->expon){
        if(lnode1->coef+lnode2->coef!=0){
            node=(Node*)malloc(sizeof(Node));
            node->next=NULL;
            node->expon=lnode1->expon;
            node->coef=lnode1->coef+lnode2->coef;
            lnode3->next=node;
            lnode3=lnode3->next;
            lnode1=lnode1->next;
            lnode2=lnode2->next;
        }
    }

}
if(lnode1){
    lnode3->next=lnode1;
}
if(lnode2){
    lnode3->next=lnode2;
}
}

```

```

    return list3;

}

int main(){
    LinkList list1,list2,list3,list4;
    Node *node1,*node2;
    list1=init();
    list2=init();
    list3=muilt(list1,list2);
    list4=add(list1,list2);
    node1=list3;
    node2=list4;
    if(!node1){
        printf("0 0\n");
    }
    if(node1&&node1->next){
        printf("%d %d",node1->next->coef,node1->next->expon);
        node1=node1->next;
    }
    while(node1&&node1->next){
        printf(" %d %d",node1->next->coef,node1->next->expon);
        node1=node1->next;
    }
    if(node1){
        printf("\n");
    }

    if(!node2){
        printf("0 0\n");
        return;
    }

    if(node2&&node2->next){
        printf("%d %d",node2->next->coef,node2->next->expon);
        node2=node2->next;
    }
    while(node2&&node2->next){
        printf(" %d %d",node2->next->coef,node2->next->expon);
        node2=node2->next;
    }
}

```