#include<stdio.h>

#include<stdlib.h>

#define cmp(x,y)(((x)>(y))?1:((x)==(y))?0:-1)

struct poly{

int expw;

int coef;

struct poly \*link;

};

typedef struct poly \*polyptr;

polyptr avail;

void createanl()

{

avail->link=NULL;

}

polyptr getnode()

{

polyptr node;

if(avail)

{

node=avail;

avail=avail->link;

}

else

{

node=(struct poly\*)malloc(sizeof(struct poly));

}

return node;

}

void attach(int coef,int expw,polyptr \*ptr)

{

polyptr temp;

temp=(struct poly\*)malloc(sizeof(struct poly));

(\*ptr)->link=temp;

temp->coef=coef;

temp->expw=expw;

temp->link=NULL;

\*ptr=temp;

}

polyptr cpadd(polyptr a,polyptr b)

{

polyptr starta,startb;

polyptr c,lastc;

int sum,done=0;

starta=a;

a=a->link;

startb=b;

b=b->link;

c=getnode();

lastc=c;

do

{

switch (cmp(a->expw,b->expw))

{

case -1:

attach(b->coef,b->expw,&lastc);

b=b->link;

break;

case 0:

if((starta==a)&&(startb==b))

{ done=1;

break;

}

else

{

sum=(a->coef)+(b->coef);

if(sum)

{

attach(sum,a->expw,&lastc);

a=a->link;

b=b->link;

}

break;

}

case 1:

attach(a->coef,a->expw,&lastc);

a=a->link;

break;

}

} while (done!=1);

for(;starta!=a;a=a->link)

{

attach(a->coef,a->expw,&lastc);

}

for(;startb!=b;b=b->link)

{

attach(b->coef,b->expw,&lastc);

}

lastc->link=c;

return c;

}

void main()

{

polyptr a,b,lasta,lastb;

a=getnode();

a->coef=0;

a->expw=0;

b=getnode();

b->coef=0;

b->expw=0;

lasta=a;

lastb=b;

int n;

printf("enter the number of elements in A:\n");

scanf("%d",&n);

for(int i=0;i<n;i++)

{

printf("Enter the coef and exp:\n");

int coef,expw;

scanf("%d %d",&coef,&expw);

attach(coef,expw,&lasta);

}

lasta->link=a;

printf("enter the number of elements in B:\n");

scanf("%d",&n);

for(int i=0;i<n;i++)

{

printf("Enter the coef and exp:\n");

int coef,expw;

scanf("%d %d",&coef,&expw);

attach(coef,expw,&lastb);

}

lastb->link=b;

polyptr c;

c=cpadd(a,b);

polyptr startc=c;

c=c->link;

for(;startc!=c;c=c->link)

{

if(c->link==startc)

{

printf("%dx^%d",c->coef,c->expw);

break;

}

printf("%dx^%d+",c->coef,c->expw);

}

}