#include <stdio.h>

#include <stdlib.h>

typedef int ElementType;

typedef struct Node \*PtrToNode;

struct Node {

ElementType Term;

ElementType Expon;

PtrToNode Next;

};

typedef PtrToNode List;

List Read();

void Print(List L);

List Add(List L1, List L2);

List Multi(List L1, List L2);

List MultiByNode(List L1, struct Node L2);

int main()

{

List L1, L2, LAdd, LMulti;

L1 = Read();

L2 = Read();

LMulti = Multi(L1, L2);

LAdd = Add(L1, L2);

Print(LMulti);

Print(LAdd);

return 0;

}

List Read(){

List L, head;

L = (List)malloc(sizeof(List));

L->Next = NULL;

head = L;

int n;

scanf("%d", &n);

if (n > 0){

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

List temp = (List)malloc(sizeof(List));

scanf("%d%d", &(temp->Term), &(temp->Expon));

temp->Next = NULL;

L->Next = temp;

L = temp;

}

}

return head;

}

void Print(List L){

List p = L->Next;

if (!p){

printf("0 0\n");

}

else{

while (p){

if (p->Next != NULL){

printf("%d %d ", p->Term, p->Expon);

}

else

{

printf("%d %d\n", p->Term, p->Expon);

}

p = p->Next;

}

}

}

List Add(List L1, List L2){

List L, head, t1, t2;

L = (List)malloc(sizeof(List));

L->Next = NULL;

head = L;

t1 = L1->Next;

t2 = L2->Next;

while (t1 != NULL&&t2 != NULL){

List temp = (List)malloc(sizeof(List));

temp->Next = NULL;

if (t1->Expon > t2->Expon){

temp->Term = t1->Term;

temp->Expon = t1->Expon;

L->Next = temp;

L = temp;

t1 = t1->Next;

}

else if (t1->Expon < t2->Expon)

{

temp->Term = t2->Term;

temp->Expon = t2->Expon;

L->Next = temp;

L = temp;

t2 = t2->Next;

}

else if (t1->Expon == t2->Expon){

//抵消

if ((t1->Term + t2->Term) == 0){

t1 = t1->Next;

t2 = t2->Next;

}

else{

temp->Term = t1->Term + t2->Term;

temp->Expon = t1->Expon;

L->Next = temp;

L = temp;

t1 = t1->Next;

t2 = t2->Next;

}

}

}

//剩下的项直接挂上去

if (t1 == NULL){

L->Next = t2;

}

else if (t2 == NULL){

L->Next = t1;

}

return head;

}

List Multi(List L1, List L2){

List L, head, t1, t2, temp;

L = (List)malloc(sizeof(List));

L->Next = NULL;

head = L;

t1 = L1->Next;

t2 = L2->Next;

if (t1 != NULL&&t2 != NULL){

temp = MultiByNode(L1, \*t2);

t2 = t2->Next;

while (t2){

List temp2 = MultiByNode(L1, \*t2);

temp = Add(temp, temp2);

t2 = t2->Next;

}

L->Next = temp->Next;

}

return head;

}

List MultiByNode(List L1, struct Node L2){

List L, head, t1;

L = (List)malloc(sizeof(List));

L->Next = NULL;

head = L;

t1 = L1->Next;

while (t1){

List temp = (List)malloc(sizeof(List));

temp->Next = NULL;

temp->Term = t1->Term\*L2.Term;

temp->Expon = t1->Expon + L2.Expon;

L->Next = temp;

L = temp;

t1 = t1->Next;

}

return head;

}