```
//S Harshini-185001058
#include<stdio.h>
#include<stdlib.h>
#include "definition.h"
#include "prototype.h"
int main()
{
 nd *hp1,*hp2,*add,*mul;
 hp1=emptylist();
 hp2=emptylist();
 add=emptylist();
 mul=emptylist();
 int ch=1;
 while(ch!=0)
 {
        printf("\nenter choice 1.add 2.multiply 3.enter 0 to exit ");
        scanf("%d",&ch);
        if(ch!=0)
        {
                printf("\nenter elements and 0 to end for the first polynomial");
                int i=9,j,k=9,l;
                while(i!=0)
                {
                scanf("%d",&i);
                scanf("%d",&j);
                if(i==0)
                break;
                insert(hp1,i,j);
                }
                printf("\nenter elements and 0 to end for the second polynomial");
                while(k!=0)
                {
                scanf("%d",&k);
                scanf("%d",&l);
                if(k==0)
                break;
                insert(hp2,k,l);
                }
                if(ch==1)
                {
                        sum(hp1,hp2,add);
                        printf("\n the resultant sum polynomial is \n");
                        display(add);
        }
        else
        {
            printf("\nthe resultant product polynomial is\n");
            multiply(hp1,hp2,mul);
        }
    }
    else
        break;
 }
    return 0;

}

/*sample input/output
enter choice 1.add 2.multiply 3.enter 0 to exit 1

enter elements and 0 to end for the first polynomial3 2
4 1
1 0
0 0

enter elements and 0 to end for the second polynomial3 1
2 0
0 0

the resultant sum polynomial is
(3 x^2) (7 x^1) (3 x^0)
enter choice 1.add 2.multiply 3.enter 0 to exit 2

enter elements and 0 to end for the first polynomial3 2
4 1
1 0
0 0

enter elements and 0 to end for the second polynomial3 1
2 0
0 0

the resultant product polynomial is
(9 x^3) (18 x^2) (11 x^1) (2 x^0)
enter choice 1.add 2.multiply 3.enter 0 to exit 0
```

```
*/


/*****************CONTENTS OF FILE STRUCTURE AND FUNCTION
DEFINITION********************\

typedef struct node
{
 int ele;
 int pow;
 struct node *next;
}nd;
nd *emptylist()
{
 nd *h;
 h=(nd*)malloc(sizeof(nd));
 h->next=NULL;
 return h;

}
void insert(nd *hd,int data,int deg)
{
 nd *p;
 p=(nd*)malloc(sizeof(nd));
 p->ele=data;
 p->pow=deg;
 p->next=hd->next;
 hd->next=p;
}
void sum(nd *hp1,nd *hp2,nd *add)
{
 nd *s,*t;
 s=hp1->next;
 t=hp2->next;
 while((s!=NULL) && (t!=NULL))
 {
  if(s->pow>t->pow)
  {
   insert(add,s->ele,s->pow);
   s=s->next;
  }
  else if(s->pow<t->pow)
  {
   insert(add,t->ele,t->pow);
   t=t->next;
  }
  else
  {
   int su;
   su=s->ele+t->ele;
   if(s!=0)
    insert(add,su,t->pow);
   s=s->next;
   t=t->next;

  }
 }
 if(s!=NULL)
 {
  while(s!=NULL)
  {
   insert(add,s->ele,s->pow);
   s=s->next;
  }

 }
 if(t!=NULL)
 {
  while(t!=NULL)
  {
   insert(add,t->ele,t->pow);
   t=t->next;
  }

 }
}
void display(nd *hd)
{
 nd *t;
 for(t=hd->next;t!=NULL;t=t->next)
 {
  printf("(%d x^%d) ",t->ele,t->pow);
 }
}
```

```
void multiply(nd *hp1,nd *hp2,nd *mul)
{
nd *n1,*t,*r,*d,*ptr3;
int coeff,deg;
for(n1=hp1->next;n1!=NULL;n1=n1->next)
{
  for(t=hp2->next;t!=NULL;t=t->next)
  {
    coeff=n1->ele*t->ele;
    deg=n1->pow+t->pow;
    insert(mul,coeff,deg);
  }
}
nd *ptr1, *ptr2,*dup;
   ptr1 = mul;
   ptr3=emptylist();
   while (ptr1 != NULL && ptr1->next != NULL)
   {
      ptr2 = ptr1;
      while (ptr2->next!= NULL)
      {
         if (ptr1->pow == ptr2->next->pow)
         {
            ptr1->ele = ptr1->ele+ ptr2->next->ele;
            dup = ptr2->next;
            ptr2->next = ptr2->next->next;
            free(dup);
         }
         else
         {
            ptr1->ele=0+ptr1->ele;
            ptr2=ptr2->next;
          }

      }
      insert(ptr3,ptr1->ele,ptr1->pow);
      ptr1 = ptr1->next;
   }
   insert(ptr3,ptr1->ele,ptr1->pow);
   nd *u;
   for(u=ptr3->next;u->next!=NULL;u=u->next)
   {
      printf("(%d x^%d) ",u->ele,u->pow);
    }
}


/*****************CONTENTS OF FILE PROTOTYPE********************\
nd *emptylist();
void insert(nd *hd,int data,int deg);
void sum(nd *hp1,nd *hp2,nd *add);
void display(nd *hd);
void multiply(nd *hp1,nd *hp2,nd *mul);
```