#include<stdio.h>

struct Polynomial

{

int coeff;

int expon;

struct Polynomial \*next;

};

struct Polynomial \*head1=NULL,\*head2=NULL,\*head3=NULL,\*neww,\*temp;

int main()

{

int coef,expo,cont;

create(&head1); // head1 will be created

create(&head2); // head2 will be created

add(head1,head2);

display(head3);

return 1;

}

void create (struct Polynomial \*\*head)

{

int coef,expo,cont;

struct Polynomial \*temp = \*head;

printf("Creation of 1st polynomial:");

while(1)

{

printf("Enter your coefficient value:");

scanf("%d",&coef);

printf("Enter your exponents:");

scanf("%d",&expo);

neww=(struct Polynomial\*)malloc(sizeof(struct Polynomial));

neww->coeff=coef;

neww->expon=expo;

neww->next=NULL;

if(\*head==NULL)

{

\*head=neww;

temp = \*head;

}

else

{

temp->next=neww;

temp = neww;

}

printf("Enter -1 to continue the loop:");

scanf("%d",&cont);

if(cont!=-1)

break;

}

}

void display(struct Polynomial \*head)

{

struct Polynomial \*temp=head;

if(temp==NULL)

printf("Empty\n");

while(temp!=NULL)

{

printf("%d x(pow)%d -->",temp->coeff,temp->expon);

temp = temp->next;

}

}

void add(struct Polynomial \*head1,struct Polynomial \*head2)

{

struct Polynomial \*h1=head1,\*h2=head2;

while(h1 && h2){

printf("Hi U I am in While");

neww=(struct Polynomial \*)malloc(sizeof(struct Polynomial));

neww->next=NULL;

if(h1->expon >h2->expon)

{

neww->coeff = h1->coeff;

neww->expon = h1->expon;

if(head3==NULL)

{

head3 = neww;

temp=head3;

}

else

{

temp->next = neww;

temp = neww;

}

h1=h1->next;

}

else if(h2->expon >h1->expon)

{

neww->coeff = h2->coeff;

neww->expon = h2->expon;

if(head3==NULL)

{

head3 = neww;

temp=head3;

}

else

{

temp->next = neww;

temp = neww;

}

h2=h2->next;

}

else if(h1->expon == h2->expon)

{

neww->coeff = h1->coeff + h2->coeff;

neww->expon = h1->expon;

if(head3==NULL)

{

head3=neww;

temp = head3;

}

else

{

temp->next = neww;

temp = neww;

}

h1=h1->next;

h2=h2->next;

}

}

while(1)

{

if(h1)

{

neww=(struct Polynomial\*)malloc(sizeof(struct Polynomial));

neww->coeff=h1->coeff;

neww->expon=h1->expon;

h1=h1->next;

neww->next=NULL;

temp->next = neww;

temp = neww;

}

else if(h2)

{

neww=(struct Polynomial\*)malloc(sizeof(struct Polynomial));

neww->coeff=h2->coeff;

neww->expon=h2->expon;

h2=h2->next;

neww->next=NULL;

temp->next = neww;

temp = neww;

}

if(h1==NULL && h2==NULL)

break;

}

}