***PROGRAM:***

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

struct node

{

int coeff;

int pow;

struct node \*next;

};

typedef struct node s1;

s1 \*start1=NULL;

s1 \*start2=NULL;

s1 \*start3=NULL;

s1 \*start4=NULL;

s1 \*last3=NULL;

s1 \*createpoly(s1\*);

s1 \*displaypoly(s1\*);

s1 \*addpoly(s1\*,s1\*,s1\*);

s1 \*subpoly(s1\*,s1\*,s1\*);

s1 \*addnode(s1\*,int,int);

int main()

{

int opt;

printf("\n\*\*\*\*\*\*\*\*\*MAIN MENU\*\*\*\*\*\*\*\*\*\*\*");

printf("\n1.Enter the first polynomial");

printf("\n2.Display the first polynomial");

printf("\n3.Enter the second polynomial");

printf("\n4.Display the second polynomial");

printf("\n5.Add the polynomial");

printf("\n6.Display the result");

printf("\n7.Subract the polynomial");

printf("\n8.Display the result");

printf("\n9.Exit");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do

{

printf("\nEnter your choice");

scanf("%d",&opt);

switch(opt)

{

case 1:

start1=createpoly(start1);

printf("\nFirst Polynomial is Created");

break;

case 2:

start1=displaypoly(start1);

printf("\nFirst Polynomial is Displayed");

break;

case 3:

start2=createpoly(start2);

printf("\nSecond Polynomial is Created");

break;

case 4:

start2=displaypoly(start2);

printf("\nSecond Polynomial is Displayed");

break;

case 5:

start3=addpoly(start1,start2,start3);

printf("\nTwo Polynomials are added");

break;

case 6:

start3=displaypoly(start3);

printf("\nResultant Polynomial is Displayed");

break;

case 7:

start4=subpoly(start1,start2,start4);

printf("\nTwo Polynomials are Subtracted");

break;

case 8:

start4=displaypoly(start4);

printf("\nResultant Polynomial is Displayed");

break;

case 9:

exit(0);

}

}while(opt!=9);

return 0;

}

s1 \*createpoly(s1 \*start)

{

s1 \*newnode,\*ptr;

int c,p;

printf("enter the coefficient:");

scanf("%d",&c);

printf("enter the power:");

scanf("%d",&p);

while(c!=-1)

{

if(start==NULL)

{

newnode=(s1\*)malloc(sizeof(s1));

newnode->coeff=c;

newnode->pow=p;

newnode->next=NULL;

start=newnode;

}

else

{

ptr=start;

while(ptr->next!=NULL)

{

ptr=ptr->next;

}

newnode=(s1\*)malloc(sizeof(s1));

newnode->coeff=c;

newnode->pow=p;

newnode->next=NULL;

ptr->next=newnode;

}

printf("Enter coeff");

scanf("%d",&c);

if(c==-1)

break;

printf("\n Enter power");

scanf("%d",&p);

}

return start;

}

s1 \*displaypoly(s1 \*start)

{

s1 \*ptr;

ptr=start;

while(ptr!=NULL)

{

printf("%dx%d",ptr->coeff,ptr->pow);

ptr=ptr->next;

}

return start;

}

s1 \*addpoly(s1 \*start1,s1 \*start2,s1 \*start3)

{

s1 \*ptr1,\*ptr2;

int sumcoeff,pow;

ptr1=start1,ptr2=start2;

while(ptr1!=NULL&&ptr2!=NULL)

{

if(ptr1->pow==ptr2->pow)

{

sumcoeff=ptr1->coeff+ptr2->coeff;

start3=addnode(start3,sumcoeff,ptr1->pow);

ptr1=ptr1->next;

ptr2=ptr2->next;

}

else if(ptr1->pow>ptr2->pow)

{

start3=addnode(start3,ptr1->coeff,ptr1->pow);

ptr1=ptr1->next;

}

else if(ptr1->pow<ptr2->pow)

{

start3=addnode(start3,ptr2->coeff,ptr2->pow);

ptr2=ptr2->next;

}

}

if(ptr1==NULL)

{

while(ptr2!=NULL)

{

start3=addnode(start3,ptr2->coeff,ptr2->pow);

ptr2=ptr2->next;

}

}

if(ptr2==NULL);

{

while(ptr1!=NULL)

{

start3=addnode(start3,ptr1->coeff,ptr1->pow);

ptr1=ptr1->next;

}

}

return start3;

}

s1 \*subpoly(s1 \*start1,s1 \*start2,s1 \*start4)

{

struct node \*ptr1,\*ptr2;

int subcoeff,pow;

ptr1=start1,ptr2=start2;

do

{

if(ptr1->pow==ptr2->pow)

{

subcoeff=ptr1->coeff-ptr2->coeff;

start4=addnode(start4,subcoeff,ptr2->pow);

ptr1=ptr1->next;

ptr2=ptr2->next;

}

else if(ptr1->pow>ptr2->pow)

{

start4=addnode(start4,ptr1->coeff,ptr1->pow);

ptr1=ptr1->next;

}

else if(ptr1->pow<ptr2->pow)

{

start4=addnode(start4,ptr2->coeff,ptr2->pow);

ptr2=ptr2->next;

}

}

while(ptr1!=NULL||ptr2!=NULL);

if(ptr1==NULL)

{

while(ptr2!=NULL)

{

start4=addnode(start4,ptr2->coeff,ptr2->pow);

ptr2=ptr2->next;

}

}

if(ptr2==NULL)

{

while(ptr1!=NULL)

{

start4=addnode(start4,ptr1->coeff,ptr1->pow);

ptr1=ptr1->next;

}

}

return start4;

}

s1 \*addnode(s1 \*start,intc,int p)

{

s1 \*ptr,\*newnode;

if(start==NULL)

{

newnode=(s1\*)malloc(sizeof(s1));

newnode->coeff=c;

newnode->pow=p;

newnode->next=NULL;

start=newnode;

}

else

{

ptr=start;

while(ptr->next!=NULL)

{

ptr=ptr->next;

}

newnode=(s1\*)malloc(sizeof(s1));

newnode->coeff=c;

newnode->pow=p;

newnode->next=NULL;

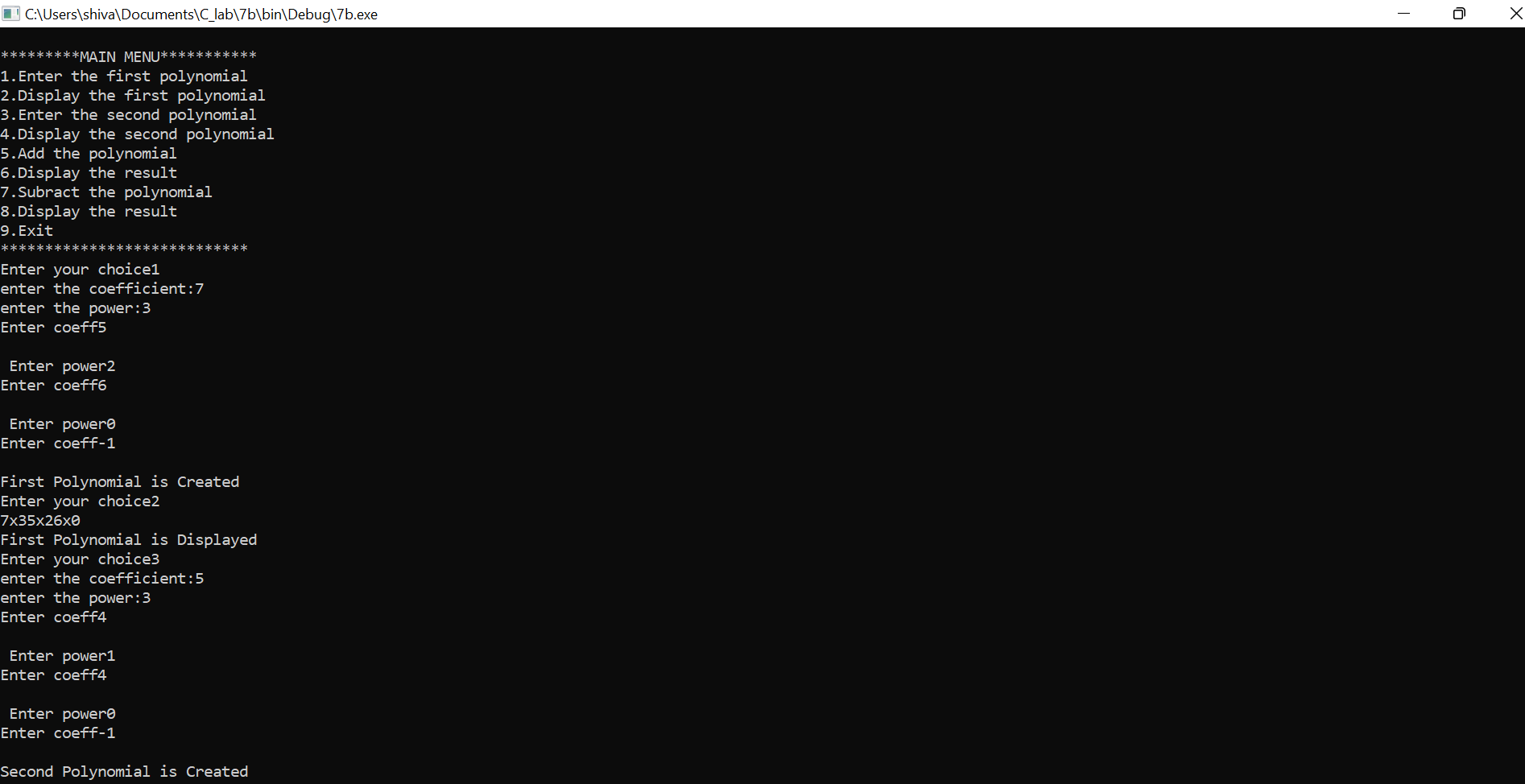
ptr->next=newnode;

}

return start;

}

***OUTPUT:***

******

