/\*addition and subtraction polynomial\*/

#include<stdio.h>

#include<stdlib.h>

struct node

{

int coeff;

int exp;

struct node\*link;

};

struct node\*header1;

struct node\*header2;

struct node\*headeradd;

struct node\*headersub;

struct node\*create\_poly(struct node\*);

struct node\*display\_poly(struct node\*);

struct node\*add\_poly(struct node\*,struct node\*,struct node\*);

struct node\*sub\_poly(struct node\*,struct node\*,struct node\*);

struct node\*add\_node(struct node\*,int ,int);

int main()

{

int choice;

while(choice!=9)

{

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

printf("1.enter the 1st poly\n2.display the 1st poly\n3.enter the 2nd poly\n4.display the 2nd poly\n5.add the polynomials\n6.display the result of addition\n7.subtraction of the polynomials\n8.display the result of subtraction\n9.exit\n");

printf("enter your choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1:header1=create\_poly(header1);

break;

case 2:header1=display\_poly(header1);

break;

case 3:header2=create\_poly(header2);

break;

case 4:header2=display\_poly(header2);

break;

case 5:headeradd=add\_poly(header1,header2,headeradd);

break;

case 6:headeradd=display\_poly(headeradd);

break;

case 7:headersub=sub\_poly(header1,header2,headersub);

break;

case 8:headersub=display\_poly(headersub);

break;

case 9:exit(0);

default:

printf("invalid choice\n");

}

}

}

struct node\*create\_poly(struct node\*header)

{

struct node\*new\_node,\*ptr;

int e,c;

printf("enter the exp num: \n");

scanf("%d",&e);

printf("enter its coefficient: \n");

scanf("%d",&c);

while(e!=-1)

{

if(header==NULL)

{

new\_node=(struct node\*)malloc(sizeof(struct node\*));

new\_node->exp=e;

new\_node->coeff=c;

new\_node->link=NULL;

header=new\_node;

}

else

{

ptr=header;

while(ptr->link!=NULL)

{

ptr=ptr->link;

}

new\_node=(struct node\*)malloc(sizeof(struct node\*));

new\_node->exp=e;

new\_node->coeff=c;

new\_node->link=NULL;

ptr->link=new\_node;

}

printf("enter the no of exp: \n");

scanf("%d",&e);

if(e==-1)

{

break;

printf("enter its coefficient: \n ");

scanf("%d",&c);

}

}

printf("polynomial is created\n");

return header;

}

struct node\*display\_poly(struct node\*header)

{

printf("the polynomial is below\n");

struct node\*ptr;

ptr=header;

while(ptr!=NULL)

{

printf("%dx%d\t\n",ptr->coeff,ptr->exp);

ptr=ptr->link;

}

return header;

}

struct node\*add\_poly(struct node\*header1,struct node\*header2,struct node\*headeradd)

{

struct node\*ptr1,\*ptr2;

int sum\_coeff;

ptr1=header1;

ptr2=header2;

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

{

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

{

sum\_coeff=ptr1->coeff+ptr2->coeff;

headeradd=add\_node(headeradd,ptr1->exp,sum\_coeff);

ptr1=ptr1->link;

ptr2=ptr2->link;

}

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

{

headeradd=add\_node(headeradd,ptr1->exp,ptr1->coeff);

ptr1=ptr1->link;

}

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

{

headeradd=add\_node(headeradd,ptr2->exp,ptr2->coeff);

ptr2=ptr2->link;

}

}

/\*if(ptr1==NULL)

{

while(ptr2!=NULL)

{

headeradd=add\_node(headeradd,ptr2->exp,ptr2->coeff);

ptr2=ptr2->link;

}

}

if(ptr2==NULL)

{

while(ptr1!=NULL)

{

headeradd=add\_node(headeradd,ptr1->exp,ptr1->coeff);

ptr1=ptr1->link;

}

}\*/

printf("addition of polynomial is done\n");

return headeradd;

}

struct node\*sub\_poly(struct node\*header1,struct node\*header2,struct node\*headersub)

{

struct node\*ptr1,\*ptr2;

int sub\_coeff;

ptr1=header1;

ptr2=header2;

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

{

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

{

sub\_coeff=ptr1->coeff-ptr2->coeff;

headersub=add\_node(headersub,ptr1->exp,sub\_coeff);

ptr1=ptr1->link;

ptr2=ptr2->link;

}

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

{

headersub=add\_node(headersub,ptr1->exp,ptr1->coeff);

ptr1=ptr1->link;

}

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

{

headersub=add\_node(headersub,ptr2->exp,ptr2->coeff);

ptr2=ptr2->link;

}

}

/\*if(ptr1==NULL)

{

while(ptr2!=NULL)

{

headersub=add\_node(headersub,ptr2->exp,ptr2->coeff);

ptr2=ptr2->link;

}

}

if(ptr2==NULL)

{

while(ptr1!=NULL)

{

headersub=add\_node(headersub,ptr1->exp,ptr1->coeff);

ptr1=ptr1->link;

}

}\*/

printf("subtraction of polynomial is done\n");

return headersub;

}

struct node\*add\_node(struct node\*HEADER,int E,int C)

{

struct node\*NEW\_NODE,\*PTR;

if(HEADER==NULL)

{

NEW\_NODE=(struct node\*)malloc(sizeof(struct node\*));

NEW\_NODE->exp=E;

NEW\_NODE->coeff=C;

NEW\_NODE->link=NULL;

HEADER=NEW\_NODE;

}

else

{

PTR=HEADER;

while(PTR->link!=NULL)

{

PTR=PTR->link;

}

NEW\_NODE=(struct node\*)malloc(sizeof(struct node\*));

NEW\_NODE->exp=E;

NEW\_NODE->coeff=C;

NEW\_NODE->link=NULL;

PTR->link=NEW\_NODE;

}

return HEADER;

}

