LAB X

```
#include <stdio.h>
#include <stdlib.h>

struct node
{
        double coeff;
        int deg;
        struct node *next;
};
typedef struct node node;

void polprint(node *);
node *polcreate(char *);
void sortpol(node *);
void insertpol(node **);
void delpol(node **);
int main()
{
   char c;
   int ch;
   node *poly;
   poly=polcreate("poly1.dat");
   if(poly==NULL)
   {
           printf("Polynomial is not created, data set is empty\n");
           return 0;
   }
   printf("The polynomial is:\n");
   polprint(poly);

   printf("Enter 1 to sort the polynomial:");
   scanf("%d",&ch);

   if(ch==1)
   {
           sortpol(poly);
   printf("After Sorting, the polynomial becomes:\n");
   polprint(poly);
   }
   else
   {
           printf("Polynomial is not sorted: stop\n");
           return 0;
   }

   printf("Enter 2 to insert a new term into the sorted polynomial:");
   scanf("%d",&ch);

   if(ch==2)
```

```
   {
           insertpol(&poly);
           printf("After insertion, the polynomial becomes:\n");
           polprint(poly);
   }

   printf("Enter 3 to delete a term from the sorted polynomial:");
   scanf("%d",&ch);

   if(ch==3)
   {
           delpol(&poly);
           printf("After deletion, the polynomial becomes:\n");
           polprint(poly);
   }


return 0;
}

void delpol(node **start)
{
        int d;
        double c;
        node *p,*q;


printf("Enter deg of the term to be deleted: ");
scanf("%d",&d);


        if((*start)->deg == d)
        {
                q=*start;
                *start=(*start)->next;
                free(q);
                return;
        }


p=*start;
q=p->next;


while(q!=NULL)
{

if(q->deg==d)
{
        p->next=q->next;
        free(q);
```

```
        return;
}
p=q;
q=q->next;
}

if(q==NULL)
{
        printf("A term with the given degree does not exist:stop\n");
        exit(1);
}


}


void insertpol(node **start)
{
        int d;
        double c;
        node *p,*q;


printf("Enter coeff and deg: ");
scanf("%lf%d",&c,&d);


        if((*start)->deg < d)
        {
                q=*start;
                *start=(node *)calloc(1,sizeof(node));
                (*start)->deg=d;
                (*start)->coeff=c;
                (*start)->next=q;
                return;
        }


p=*start;
q=p->next;


while(q!=NULL)
{

if(p->deg==d)
{
        printf("A term with the given degree already exists: stop\n");
        exit(1);
}
```

```
if(q->deg<d)
{
        p->next=(node *)calloc(1,sizeof(node));
        p=p->next;
        p->coeff=c;
        p->deg=d;
        p->next=q;
        return;
}
p=q;
q=q->next;
}

if(p->deg==d)
{
        printf("A term with the given degree already exists\n");
        exit(1);
}
        p->next=(node *)calloc(1,sizeof(node));
        p=p->next;
        p->coeff=c;
        p->deg=d;
        p->next=NULL;


}



node *polcreate(char *filename)
{
  int d;
  double c;
  FILE *inpol;
  node *head,*p;

  inpol=fopen(filename,"r");
  if(inpol==NULL)
  {printf("Error in opening %s\n",filename);
   exit(1);
  }

    fscanf(inpol,"%lf%d",&c,&d);
    if(d==-1)
    {
            return NULL;
    }
    head=(node *)calloc(1,sizeof(node));
    p=head;
    p->coeff=c;
    p->deg=d;
```

```
    p->next=NULL;

    fscanf(inpol,"%lf%d",&c,&d);
    while(d !=-1)
    {
            p->next=(node *)calloc(1,sizeof(node));
            p=p->next;
            p->coeff=c;
            p->deg=d;
            fscanf(inpol,"%lf%d",&c,&d);

    }
            p->next=NULL;


   return head;
}

void sortpol(node *start)
{
        int c;
        double d;
        node *p,*q;

        p=start;
        while(p->next !=NULL)
        {
                q=p->next;
                while(q!=NULL)
                {
                        if(p->deg<q->deg)
                       {
                        d=p->deg;
                        c=p->coeff;
                        p->deg=q->deg;
                        p->coeff=q->coeff;
                        q->deg=d;
                        q->coeff=c;
                        }
                q=q->next;
                }
                p=p->next;
        }
}



void polprint(node *p)
{
        if(p->deg==0)
        {
```

```
                printf("%0.2lf\n",p->coeff);
        }
        else
        {
                printf("%0.2lf x^%d",p->coeff,p->deg);
        }

p=p->next;

while(p!=NULL)
{
        if(p->deg==0)
        {
                printf("%+0.2lf ",p->coeff);
        }
        else
        {
                printf("%+0.2lf x^%d ",p->coeff,p->deg);
        }

        p=p->next;

}
printf("\n");

}
```