Design, Develop and Implement a Program in C for the following operations on Singly Circular
Linked List (SCLL) with header nodes
a. Represent and Evaluate a Polynomial P(x,y,z) = 6x2y2z-4yz5+3x3yz+2xy5z-2xyz3
b. Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z)
Support the program with appropriate functions for each of the above operations

```
#include<stdio.h>
#include<conio.h>
#include<alloc.h>
#include<process.h>
#include<math.h>

 struct node
{
 float cf;
int px,py,pz;
int flag;
struct node *link;
};
typedef struct node *NODE;


NODE  getnode()
{
        NODE  x;
        x=(NODE) malloc (sizeof(struct node));
        if(x==NULL)
        {
                printf("out of memory\n");
                exit(0);
        }
        return x;
}

NODE insert_rear(float cf,float  x , float  y, float  z,NODE head)
{
        NODE temp,cur;
        temp=getnode();
        temp->cf=cf;
        temp->px=x;
        temp->py=y;
        temp->pz=z;
        temp->flag=0;
        cur=head->link;
        while(cur->link!=head)
        {
                cur=cur->link;
        }
        cur->link=temp;
        temp->link=head;
        return head;
}


void display(NODE head)
{
        NODE  temp;
        if(head->link==head)
        {
                printf("polynominal doesn't exsits\n");
                return;
        }
        temp=head->link;
        while(temp!=head)
        {
                printf("+ %  5.2fx^%dy^%dz^%d",temp->cf,temp->px,temp->py,temp->pz);
                temp=temp->link;
        }
        printf("\n");
        }
NODE  add_poly(NODE h1,NODE h2, NODE h3)
{
        NODE p1,p2;
        int x1,x2,y1,y2,z1,z2,cf1,cf2,cf;
        p1=h1->link;
        while(p1!=h1)
{
                x1=p1->px;
                y1=p1->py;
                z1=p1->pz;
                cf1=p1->cf;
                p2=h2->link;
                while(p2!=h2)
{
 x2=p2->px;
y2=p2->py;
z2=p2->pz;
cf2=p2->cf;
if(x1==x2 &&  y1==y2 && z1==z2)
        break;
        p2=p2->link;
}
        if(p2!=h2)
{
        cf=cf1+cf2;
        p2->flag=1;
        if(cf!=0)
        h3=insert_rear(cf,x1,y1,z1,h3);
        }
else
h3=insert_rear(cf1,x1,y1,z1,h3);
p1=p1->link;
}

p2=h2->link;
while(p2!=h2)
{
        if(p2->flag==0)
{
h3=insert_rear(p2->cf,p2->px,p2->py,p2->pz,h3);
}
p2=p2->link;
}
return h3;
}

NODE read_poly(NODE head)
{
                int i;
        int px,py,pz;
        float cf;
printf("enter the coeffecient as -999 to end the polynominal\n");
for(i=1;;i++)
{
printf("enter the %d term\n",i);
printf("coeff=");
scanf("%f",&cf);
if(cf==-999) break;
printf("pow x=");
scanf("%d",&px);
printf("pow y=");
scanf("%d",&py);
printf("pow z=");
scanf("%d",&pz);

head=insert_rear(cf,px,py,pz,head);
}
return head;
}


void polysum()
{
NODE h1,h2,h3;
h1=getnode();
h2=getnode();
h3=getnode();
```

```
h1->link=h1;
h2->link=h2;
h3->link=h3;
printf("enter the first polynominal\n");
h1=read_poly(h1);
printf("enter the second polynominal\n");
h2=read_poly(h2);
h3=add_poly(h1,h2,h3);
printf(" the first polynominal is\n");
display(h1);
printf("second polynominal is\n");
display(h2);
printf("the sum of two polynominal is\n");
display(h3);
}
void represent_evaluate()
{
NODE e1,temp;
int x,y,z;
float sum=0.0;
e1=getnode();
e1->link=e1;
printf("enter the polynominal\n");
e1=read_poly(e1);
printf("polynominal i s \n");
display(e1);
printf("enter the values of coefficient\n");
scanf("%d%d%d",&x,&y,&z);
  if(e1==NULL)
  {
  printf("list is empty");

  }
  else
  {
                temp=e1->link;
                while(temp!=e1)
                {

                        sum+=temp->cf*pow(x,temp->px)*pow(y,temp->py)*pow(z,temp-
>pz);
                        temp=temp->link;
                }
  //    sum+=temp->cf*pow(x,temp->px)*pow(y,temp->py)*pow(z,temp->pz);
        printf("the total sum is %f\n",sum);
 }
return;
 }
void main()
{
 int choice;
        clrscr();
        while(1)
        {
                printf("\n\n\n\t1.represent and evaluate...\t2.ADD TWO poly..\t3.Exit...");
                printf("\n\n\n\tEnter Your Choice: ");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1: represent_evaluate();break;
                        case 2:polysum();break;
                        case 3:exit(0);
                        default: printf("\n\n\n\tEnter proper Choice....");
                }
        }

}
```