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) = $6x^2y^2z-4yz^5+3x^3yz+2xy^5z-2xyz^3$

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

```c
#include <stdio.h>
#include <stdlib.h>
#include<math.h>
typedef struct polynomial
{
   float coeff;
   int x,y,z;
   struct polynomial *next;
}poly;
poly *p1,*p2,*p3;
poly* readpoly()
{
   poly *temp=(poly*)malloc(sizeof(poly));
   printf("\nEnter coeff:");
   scanf("%f",&temp->coeff);
   printf("Enter x expon:");
   scanf("%d",&temp->x);
   printf("Enter y expon:");
   scanf("%d",&temp->y);
   printf("Enter z expon:");
   scanf("%d",&temp->z);
   return temp;
}
poly* create()
{
   int n,i;
   printf("\nEnter no. of terms:");
   scanf("%d",&n);
   poly *temp=(poly*)malloc(sizeof(poly)),*t1=temp;
   for(i=0;i<n;i++,t1=t1->next)
     t1->next=readpoly();
   t1->next=temp;
   return temp;
```

```c
}
void evaluate()
{
    float sum=0;
    int x,y,z;
    poly *t=p1->next;
    printf("\nEnter x,y&z:\n");
    scanf("%d",&x);
    scanf("%d",&y);
    scanf("%d",&z);
    while(t!=p1)
    {
        sum+=t->coeff*pow(x,t->x)*pow(y,t->y)*pow(z,t->z);
        t=t->next;
    }
    printf("\nSum=%f",sum);
}
void display(poly *p)
{
    poly *t=p->next;
    while(t!=p)
    {
        if(t!=p->next&&t->coeff>0)
            putchar('+');
        printf("%.1fx^%dy^%dz^%d",t->coeff,t->x,t->y,t->z);
        t=t->next;
    }
}
poly* attach(float coeff,int x,int y,int z,poly *p)
{
    poly *t=(poly*)malloc(sizeof(poly));
    t->coeff=coeff;
    t->x=x;
    t->y=y;
    t->z=z;
    p->next=t;
    return t;
}
poly* add()
{
    printf("\nPolynomial1:\n");
    p1=create();
    printf("\nPolynomial2:\n");
```

```c
    p2=create();
    int flag;
    poly *t1=p1->next,*t2=p2->next,*t3;
    p3=(poly*)malloc(sizeof(poly));
    t3=p3;
    while(t1!=p1&&t2!=p2)
    {
       if(t1->x>t2->x)
         flag=1;
       else if(t1->y<t2->y)
         flag=-1;
       else if(t1->z==t2->z)
         flag=0;
       switch(flag)
       {
          case 0:t3=attach(t1->coeff+t2->coeff,t1->x,t1->y,t1->z,t3);
                 t1=t1->next;
                 t2=t2->next;
                 break;
          case 1:t3=attach(t1->coeff,t1->x,t1->y,t1->z,t3);
                 t1=t1->next;
                 break;
          case -1:t3=attach(t2->coeff,t2->x,t2->y,t2->z,t3);
                  t2=t2->next;
                  break;
       }
    }
    for(;t1!=p1;t1=t1->next)
      t3=attach(t1->coeff,t1->x,t1->y,t1->z,t3);
    for(;t2!=p2;t2=t2->next)
      t3=attach(t2->coeff,t2->x,t2->y,t2->z,t3);
    t3->next=p3;
    return p3;
}
int main()
{
  int ch;
  printf("\n1.Represent and evaluate polynomial\n2.Add 2 polynomials\n3.Exit\nEnter choice:");
  scanf("%d",&ch);
  switch(ch)
  {
     case 1:p1=create();
            display(p1);
```

```c
            evaluate();
            break;
    case 2:p3=add();
            printf("\nPolynomial1:\n");
            display(p1);
            printf("\nPolynomial2:\n");
            display(p2);
            printf("\nP1+P2:\n");
            display(p3);
            break;
    case 3:exit(0);
    default:printf("\nInvalid choice...!");
  }
 return 0;

}
```