```
/*9.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 <stdlib.h>
#include <math.h>

struct Term
{
    int exp[3];
    int coef;
    struct Term* link;
};

typedef struct Term* Polynomial;

Polynomial getnode()
{
    Polynomial x;
    x = (Polynomial) malloc(sizeof(struct Term));
    return x;
}

Polynomial attach(Polynomial head, int exp[3], int coef)
{
    int i;
    Polynomial temp, cur;
    temp = getnode();
    for(i=0 ; i<3 ; i++)
        temp->exp[i] = exp[i];
    temp->coef = coef;
    cur = head->link;
    while(cur->link != head)
    {
        cur = cur->link;
    }
    cur->link = temp;
    temp->link = head;
    return head;
}

Polynomial read_poly(Polynomial head)
{
    int nterms,coef, exp[3], i;
    printf("\nEnter the No. of Terms for the polynomial: \n");
    scanf("%d", &nterms);
    for(i = 1; i <= nterms; i++)
    {
        printf("Enter the co-efficient of %d term: \n", i);
        scanf("%d", &coef);
        printf("Enter the exponents of %d term...\n",i);
        printf("Enter the exponent of x: \n", i);
        scanf("%d", &exp[0]);
        printf("Enter the exponent of y: \n", i);
        scanf("%d", &exp[1]);
        printf("Enter the exponent of z: \n", i);
        scanf("%d", &exp[2]);
        head= attach(head, exp, coef);
    }
    return head;
}
 void display(Polynomial head)
{
    Polynomial temp;
    if(head->link == head)
    {
        printf("\nPolynomial does not exist");
        return;
    }
    temp = head->link;
    while(temp != head)
    {
        if(temp->coef!=0)
        {
            printf("%d",temp->coef);
            if(temp->exp[0]!=0)
                printf("x^%d ",temp->exp[0]);
            if(temp->exp[1]!=0)
                printf("y^%d ",temp->exp[1]);
            if(temp->exp[2]!=0)
                printf("z^%d",temp->exp[2]);
        }
            temp = temp->link;
            if(temp != head)
                printf(" + ");
    }
}
int compare(int a[3],int b[3])
{
    if(a[0] == b[0])
    {
        if(a[1] == b[1])
        {
            if(a[2] == b[2])
                return 0;
            if(a[2] > b[2])
                return 1;
            if(a[2] < b[2])
                return -1;
        }
        if(a[1] > b[1])
            return 1;
        if(a[1] < b[1])
            return -1;
    }
    if(a[0] > b[0])
        return 1;
    if(a[0] < b[0])
```

```
        return -1;
}

Polynomial poly_add(Polynomial head1, Polynomial head2, Polynomial head3)
{
    Polynomial a,b;
    int coeff;
    a = head1->link;
    b = head2->link;
    while(a != head1 && b != head2)
    {
        switch(compare(a->exp, b->exp))
        {
            case 0:
                    coeff = a->coef + b->coef;
                    if(coeff != 0)
                    head3 = attach(head3, a->exp, coeff);
                    a = a->link;
                    b = b->link;
                    break;
            case 1:
                    head3 = attach(head3, a->exp, a->coef);
                    a=a->link;
                    break;
            case -1:
                    head3 = attach(head3, b->exp, b->coef);
                    b = b->link;
                    break;

        }
    }
    while(a != head1)
    {
        head3 = attach(head3, a->exp, a->coef);
        a = a->link;
    }
    while(b != head2)
    {
        head3 = attach(head3, b->exp, b->coef);
        b = b->link;
    }
    return head3;
}
void evalPoly(Polynomial head)
{
    long sum = 0;
    int x, y, z;
    Polynomial p;
    printf("\n\nEnter the value of x, y and z\n ");
    scanf("%d%d%d",&x, &y, &z);
    p = head->link;
    while(p!= head)
    {
        sum += (p->coef * pow(x, p->exp[0]) * pow(y, p->exp[1]) * pow(z, 
p->exp[2]));
        p = p->link;
    }
    printf(" Result of P(x,y,z) is : %ld\n", sum);
```

```
}
void main()
{
    int  ch;
      char ele;
    Polynomial polyA, polyB, polyC, poly;
    poly = (Polynomial)malloc(sizeof(struct Term));
    polyA = (Polynomial)malloc(sizeof(struct Term));
    polyB = (Polynomial)malloc(sizeof(struct Term));
    polyC = (Polynomial)malloc(sizeof(struct Term));
    polyA->link = polyA;
    polyB->link = polyB;
    polyC->link = polyC;
    poly->link = poly;
    while (1)
      {
           printf("\n\n\n********** Singly Circular Linked List (SCLL)
Operation Menu**********\n\n");
           printf("1. Represent and Evaluate a Polynomial P(x,y,z)\n");
           printf("2. Sum of two polynomials POLY1(x,y,z) and
POLY2(x,y,z)\n");
           printf("4. Exit\n");
           printf("Enter your choice:\n");
           scanf("%d", &ch);
           switch (ch)
           {
            case 1: printf("Enter the polynomial P(x,y,z) to Represent
and Evaluate\n");
                    poly = read_poly(poly);
                    printf("Polynomial P(x,y,z) is:");
                    display(poly);
                    evalPoly(poly);
                    break;
            case 2: printf("Enter the Polynomial POLY1(x,y,z)\n\n");
                    polyA = read_poly(polyA);
                    printf("Enter the Polynomial POLY2(x,y,z)\n\n");
                    polyB = read_poly(polyB);
                    printf("\n\nPolynomial POLY1(x,y,z) is: ");
                    display(polyA);

                    printf("\nPolynomial POLY2(x,y,z) is: ");
                    display(polyB);

                    polyC = poly_add(polyA, polyB, polyC);

                    printf("\n\nSum of two polynomials POLY1(x,y,z) and
POLY2(x,y,z) is : ");
                    display(polyC);
                    break;
            case 3: exit(0);
            default: printf("Enter a Valid Choice\n");
                    break;
           }
      }
}
```