```c
/*
DS9:Design, Develop and Implement a Program in C for the following operations on
    Singly Circular Linked List (SCLL) with header nodes.
    a. Represent & Evaluate a Polynomial P(x,y,z) = 6x2y2z-4yz5+3x3yz+2xy5z-2xyz3
    b. Find the sum of two polynomials POLY1(x,y,z) & 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>

#define COMPARE(x, y) ((x == y)? 0 : (x > y)? 1 : -1)

/*
coef:Coefficient of each term.
xexp,yexp,zexp: Powers of x,y,z respectively.

The following functions are used in the program:
get node() function
NODE attach() function
NODE readpoly() function
void display() function
int polyevaluate() function
NODE polysum() function
main() function
*/

struct node
{
    int coef;
    int xexp, yexp, zexp;
    struct node *link;
};

typedef struct node *NODE;           //struct node is renamed to NODE

//get node() function is used to allocate memeory to each node that is created.
NODE getnode()
{
    NODE x;
    x = (NODE) malloc(sizeof(struct node));
    return x;
}

//temp is pointer variable that is used to add new node or display node.
NODE attach(int coef, int xexp, int yexp, int zexp, NODE head)
{
    NODE temp;
    NODE cur;   //cur is pointer variable that represents the current node

    temp = getnode();

    temp->coef = coef;
    temp->xexp = xexp;
    temp->yexp = yexp;
    temp->zexp = zexp;

    cur = head->link;                    //Points to the first node.

    while(cur->link != head)            //till the end of circular list
        {
            cur = cur->link;           //keep on moving forward.
        }
```

```
        cur->link = temp;          //attach the new node to the circular list.
        temp->link = head;        //link the attached node back to head.
        return head;
}

/*
Reads the Polynomial.
Takes the input of Number of terms, Co-efficient and powers of X,Y,Z)
*/
NODE readpoly(NODE head)
{
    int i, j;
    int n;
    int coef, xexp, yexp, zexp;

    printf("\n Enter the no of terms in the polynomial: ");
    scanf("%d",&n);

    for(i=1; i<=n; i++)                    //from 1st term to nth term
        {
            printf("\n Enter the %d term: ",i);

            printf("\n Coef =  ");
            scanf("%d", &coef);

            printf("\n Enter Pow(x) Pow(y) and Pow(z): ");
            scanf("%d", &xexp);
            scanf("%d", &yexp);
            scanf("%d", &zexp);

            head = attach(coef, xexp, yexp, zexp, head); //Put all this in a node
        }

        return head; //All the details will be in the head node.
}

void display(NODE head)
{
    NODE temp;

    if(head->link == head)         //if there is no circular linked list
        {
            printf("\n Polynomial does not exist");
            return;
        }

        temp = head->link;             //Initialise from first node

        while(temp != head)           //till the end of circular list
            {
                printf("%dx^%dy^%dz^%d", temp->coef,temp->xexp,temp->yexp,temp->zexp);
                temp = temp->link;         //Keep moving to next term

                if(temp != head)
                    printf(" + ");  //print '+' after printing every term
            }
}

// This part evaluates the polynomial and returns the sum.
// pow is defined under math.h
int polyevaluate(NODE head)
{
    NODE poly;            //here poly is same as temp. Pointer varaible.
    int x,y,z;
    int sum = 0;
```

```
    printf("\n Enter the value of x,y and z: ");
    scanf("%d%d%d", &x,&y,&z);

    poly = head->link;          //

    while(poly != head)                    //Till the end of linked list
        {
            sum += poly->coef * pow(x,poly->xexp)* pow(y,poly->yexp) * pow(z,poly->zexp);
            poly = poly->link;             //move to next term
        }
        return sum;
}

/*
This part calculates the sum of two polynomials and returns the sum.
sum can be calculated only when exponentials (powers) of x,y,z are same
in both the terms.
If they are same, sum is calculated by adding their coefficients.
That sum is stored in new polynomial represented by head3.
*/
NODE polysum(NODE head1, NODE head2, NODE head3)
{
    NODE a,b;
    int coef;

    a = head1->link;                    //a represents first polynomial
    b = head2->link;                   //b represents second polynomial

    while(a!=head1 && b!=head2) //Till the last nodes in both polynomials
    {
        while(1)
            {
                if(a->xexp == b->xexp && a->yexp == b->yexp && a->zexp == b->zexp)
                {
                    coef = a->coef + b->coef;       //add coefficients
                    head3 = attach(coef, a->xexp, a->yexp, a->zexp, head3);
                    a = a->link;              //Move to next term in 1st polynomial
                    b = b->link;             //Move to next term in 1st polynomial
                    break;
                }

                if(a->xexp!=0 || b->xexp!=0) //if pow(x) is not equal to zero
                    {
                        switch(COMPARE(a->xexp, b->xexp)) //compare pow(x) in both poly
                        {
                            case -1: head3 = attach(b->coef, b->xexp, b->yexp, b->zexp,
head3);
                                     b = b->link;
                                     break;

                            case 0 : if(a->yexp > b->yexp)
                                     {
                                       head3 = attach(a->coef, a->xexp, a->yexp, a->zexp,
head3);
                                       a = a->link;
                                       break;
                                     }

                                    else if(a->yexp < b->yexp)
                                        {
                                            head3 = attach(b->coef, b->xexp, b->yexp, b->
zexp, head3);
                                            b = b->link;
                                            break;
                                        }
```

```
                                    else if(a->zexp > b->zexp)
                                        {
                                            head3 = attach(a->coef, a->xexp, a->yexp, a->
zexp, head3);
                                            a = a->link;
                                            break;
                                        }

                                    else if(a->zexp < b->zexp)
                                        {
                                            head3 = attach(b->coef, b->xexp, b->yexp, b->
zexp, head3);
                                            b = b->link;
                                            break;
                                        }

                                case 1 : head3 = attach(a->coef, a->xexp, a->yexp, a->zexp,
head3);
                                         a = a->link;
                                         break;
                            }
                            break;

                        } //end of comparing power of x in polynomial.

                    if(a->yexp!=0 || b->yexp!=0)
                     {
                        switch(COMPARE(a->yexp, b->yexp))
                        {
                            case -1 :   head3 = attach(b->coef, b->xexp, b->yexp, b->zexp,
head3);
                                        b = b->link;
                                        break;

                            case 0 :   if(a->zexp > b->zexp)
                                {
                                    head3 = attach(a->coef, a->xexp, a->yexp, a->zexp, head3
);
                                    a = a->link;
                                    break;
                                }

                                    else if(a->zexp < b->zexp)
                                    {
                                        head3 = attach(b->coef, b->xexp, b->yexp, b->zexp
, head3);
                                        b = b->link;
                                        break;
                                    }

                            case 1 :    head3 = attach(a->coef, a->xexp, a->yexp, a->zexp,
head3);
                                        a = a->link;
                                        break;
                        }
                                        break;

                    } //end of comparing power of y in polynomial.

                    if(a->zexp!=0 || b->zexp!=0)
                        {
                            switch(COMPARE(a->zexp, b->zexp))
                            {
                                case -1: head3 = attach(b->coef,b->xexp,b->yexp,b->zexp,head3
);
                                         b = b->link;
```

```
                                    break;

                            case 1 : head3 = attach(a->coef, a->xexp, a->yexp, a->zexp,
head3);
                                    a = a->link;
                                    break;
                        }
                            break;

                    }//end of comparing power of z in polynomial.
            }//end of while loop

    }//end of while condition

    while(a!= head1) //till the end of expression in 1st polynomial.
        {
            head3 = attach(a->coef,a->xexp,a->yexp,a->zexp,head3);
            a = a->link;
        }

    while(b!= head2) //till the end of expression in 2nd polynomial.
        {
            head3 = attach(b->coef,b->xexp,b->yexp,b->zexp,head3);
            b = b->link;
        }

    return head3; //return head3 that stores sum polynomial.
}

void main()
{
    NODE head, head1, head2, head3;
    int res, ch;

    head =  getnode();                      // For polynomial evalaution

    head1 = getnode();                    // To hold POLY 1
    head2 = getnode();                   // To hold POLY 2
    head3 = getnode();                  // To hold POLYSUM

//initially only head node will be present in Linked List.(Empty Linked List)
    head->link=head;
    head1->link=head1;
    head2->link=head2;
    head3->link= head3;

    while(1)
        {
            printf("\n 1 - Represent and Evaluate a Polynomial P(x,y,z)");
            printf("\n 2 - Find the sum of two polynomials POLY(x,y,z)");

            printf("\n Enter your choice:");
            scanf("%d",&ch);

            switch(ch)
            {
                case 1: printf("Polynomial evaluation P(x,y,z)\n");
                        head = readpoly(head);

                        printf("\n Representation of Polynomial for evaluation: \n");
                        display(head);

                        res = polyevaluate(head);
                        printf("\n Result of polynomial evaluation is: %d\n", res);
                        break;
```

```
                case 2: printf("Enter the POLY1(x,y,z):\n");
                        head1 = readpoly(head1);
                        printf("\n Polynomial 1 is:\n");
                        display(head1);

                        printf("\n Enter the POLY2(x,y,z):\n");
                        head2 = readpoly(head2);
                        printf("\n Polynomial 2 is:\n");
                        display(head2);

                        printf("\n Polynomial sum is:\n");
                        head3 = polysum(head1,head2,head3);
                        display(head3);
                        break;

                case 3: exit(0);

            }//end of switch case

        }//end of while loop

}//end of main function
```

OUTPUT 1:

```
1 - Represent and Evaluate a Polynomial P(x,y,z)
2 - Find the sum of two polynomials POLY(x,y,z)
Enter your choice:1
Polynomial evaluation P(x,y,z)

Enter the no of terms in the polynomial: 5

Enter the 1 term:
Coef =  6

Enter Pow(x) Pow(y) and Pow(z): 2 2 1

Enter the 2 term:
Coef =  -4

Enter Pow(x) Pow(y) and Pow(z): 0 1 5

Enter the 3 term:
Coef =  3

Enter Pow(x) Pow(y) and Pow(z): 3 1 1

Enter the 4 term:
Coef =  2

Enter Pow(x) Pow(y) and Pow(z): 1 5 1

Enter the 5 term:
Coef =  -2

Enter Pow(x) Pow(y) and Pow(z): 1 1 3

Representation of Polynomial for evaluation:
6x^2y^2z^1 + -4x^0y^1z^5 + 3x^3y^1z^1 + 2x^1y^5z^1 + -2x^1y^1z^3
Enter the value of x,y and z: 1 1 1

Result of polynomial evaluation is: 5

1 - Represent and Evaluate a Polynomial P(x,y,z)
2 - Find the sum of two polynomials POLY(x,y,z)
 Enter your choice:2
 Enter the POLY1(x,y,z):

 Enter the no of terms in the polynomial: 3

 Enter the 1 term:
 Coef =  4
```

```
 Enter Pow(x) Pow(y) and Pow(z): 3 3 3

 Enter the 2 term:
 Coef =  3

 Enter Pow(x) Pow(y) and Pow(z): 2 2 2

 Enter the 3 term:
 Coef =  2

 Enter Pow(x) Pow(y) and Pow(z): 1 1 1

 Polynomial 1 is:
4x^3y^3z^3 + 3x^2y^2z^2 + 2x^1y^1z^1
 Enter the POLY2(x,y,z):

 Enter the no of terms in the polynomial: 3

 Enter the 1 term:
 Coef =  8

 Enter Pow(x) Pow(y) and Pow(z): 3 3 3

 Enter the 2 term:
 Coef =  5

 Enter Pow(x) Pow(y) and Pow(z): 2 2 2

 Enter the 3 term:
 Coef =  4

 Enter Pow(x) Pow(y) and Pow(z): 1 1 1

 Polynomial 2 is:
8x^3y^3z^3 + 5x^2y^2z^2 + 4x^1y^1z^1

 Polynomial sum is:
12x^3y^3z^3 + 8x^2y^2z^2 + 6x^1y^1z^1

**********************************************************************

OUTPUT 2:

1 - Represent and Evaluate a Polynomial P(x,y,z)
2 - Find the sum of two polynomials POLY(x,y,z)
Enter your choice:1
Polynomial evaluation P(x,y,z)

Enter the no of terms in the polynomial: 3
```

```
Enter the 1 term:
Coef =  2

Enter Pow(x) Pow(y) and Pow(z): 3 2 1

Enter the 2 term:
Coef =  2

Enter Pow(x) Pow(y) and Pow(z): 2 3 1

Enter the 3 term:
Coef =  2

Enter Pow(x) Pow(y) and Pow(z): 1 2 3

Representation of Polynomial for evaluation:
2x^3y^2z^1 + 2x^2y^3z^1 + 2x^1y^2z^3
Enter the value of x,y and z: 1 2 1

Result of polynomial evaluation is: 32

1 - Represent and Evaluate a Polynomial P(x,y,z)
2 - Find the sum of two polynomials POLY(x,y,z)
Enter your choice:2
Enter the POLY1(x,y,z):

Enter the no of terms in the polynomial: 3

Enter the 1 term:
Coef =  2

Enter Pow(x) Pow(y) and Pow(z): 1 2 3

Enter the 2 term:
Coef =  4

Enter Pow(x) Pow(y) and Pow(z): 2 3 1

Enter the 3 term:
Coef =  -6

Enter Pow(x) Pow(y) and Pow(z): 3 2 1

Polynomial 1 is:
2x^1y^2z^3 + 4x^2y^3z^1 + -6x^3y^2z^1
Enter the POLY2(x,y,z):

Enter the no of terms in the polynomial: 3

Enter the 1 term:
```

```
Coef =  1

Enter Pow(x) Pow(y) and Pow(z): 3 2 1

Enter the 2 term:
Coef =  -2

Enter Pow(x) Pow(y) and Pow(z): 1 2 3

Enter the 3 term:
Coef =  3

Enter Pow(x) Pow(y) and Pow(z): 2 3 1

Polynomial 2 is:
1x^3y^2z^1 + -2x^1y^2z^3 + 3x^2y^3z^1

Polynomial sum is:
0x^1y^2z^3 + 7x^2y^3z^1 + -5x^3y^2z^1
```