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

struct node
{

   int cf, px, py, pz,flag;
   struct node *link;
};

typedef struct node NODE;

NODE *head,*h1,*h2,*h3;

NODE* getnode()
{
   NODE *temp = (NODE*)malloc(sizeof(NODE));
   temp->link=NULL;
   temp->flag=0;
   printf(" coefficient and exponents of X, Y, Z\n");
   scanf("%d%d%d%d",&temp->cf,&temp->px,&temp->py,&temp->pz);
   return temp;
}

void insert_rear(NODE* h)
{
   NODE *temp = getnode(),*temp1;

   if(h->link==h)
   {
```

```
        h->link=temp;
        temp->link=h;
    }
    else
    {

        for(temp1=h->link;temp1->link!=h;temp1=temp1->link)
        {

        }
        temp1->link=temp;
        temp->link=h;
    }

}

void attach(int cf,int px,int py,int pz)
{
    NODE *temp = (NODE*)malloc(sizeof(NODE));
    temp->link=NULL;
    temp->cf=cf;
    temp->px=px;
    temp->py=py;
    temp->pz=pz;

    NODE *temp1;

    if(h3->link==h3)
    {
        h3->link=temp;
```

```
        temp->link=h3;
    }
    else
    {

        for(temp1=h3->link;temp1->link!=h3;temp1=temp1->link)
        {

        }
        temp1->link=temp;
        temp->link=h3;
    }
}

void display(NODE* h)
{
    NODE *temp;

    if(h->link==h)
    {

      printf("ZERO POLYNOMIAL\n");
    }
    else
    {

        for(temp=h->link;temp->link!=h;temp=temp->link)
        {
            printf("%dX^%dY^%dZ^%d+",temp->cf,temp->px,temp->py,temp->pz);
        }
        printf("%dX^%dY^%dZ^%d\n",temp->cf,temp->px,temp->py,temp->pz);
```

```
    }

}

void evaluate()
{
    NODE *temp;
    int X, Y, Z;
   int sum=0;
    if(head->link==head)
    {

      printf("ZERO POLYNOMIAL\n");
    }
    else
    {
        printf("Enter the value of variables X,Y and Z");
        scanf("%d%d%d",&X,&Y,&Z);

        for(temp=head->link;temp!=head;temp=temp->link)
        {
            sum+=temp->cf*pow(X,temp->px)*pow(Y,temp->py)*pow(Z,temp->pz);

        }
        printf("EVALUATION RESULT: %d",sum);

    }

}
```

```
NODE* read(NODE* h)
{
  int i, N;
    h = (NODE*)malloc(sizeof(NODE));
    h->link=h;

    printf("Enter the number of terms\n");
    scanf("%d",&N);
    for(i=1;i<=N;i++)
    {

        printf("Enter the %d term",i);
        insert_rear(h);
    }
    return h;

}



void  add_poly()
{
   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)
          attach(cf,x1,y1,z1);
     }
 else
          attach(cf1,x1,y1,z1);
 p1=p1->link;
}

p2=h2->link;
while(p2!=h2)
 {
   if(p2->flag==0)
    {
```

```
            attach(p2->cf,p2->px,p2->py,p2->pz);
          }
        p2=p2->link;
      }
}

void main()
{
   int choice;
   while(1)
   {
      printf("\n 1->Represent and Evaluate\n 2->Addition of two polynomial Eqn.\n 3->Exit\n");
      printf("Enter Your choice\n");
      scanf("%d",&choice);
      switch(choice)
      {

      case 1:printf("Enter the polynomial Equation\n");
            head=read(head);
            printf("Entered Polynomial Equation is P(X,Y,Z) = ");
            display(head);
            evaluate();
          break;
      case 2:printf("Enter the First polynomial Equation\n");
            h1=read(h1);
            printf("Enter the Second polynomial Equation\n");
            h2=read(h2);
            printf("\n Entered First Polynomial Equation is POLY1(X,Y,Z) = ");
            display(h1);
            printf("\n Entered Second Polynomial Equation is POLY2(X,Y,Z) = ");
            display(h2);
```

```
printf("_____
_____\n");
            printf("Addition of Two Polynomial Equation is POLYSUM(X,Y,Z) = ");
            h3=(NODE*)malloc(sizeof(NODE));
            h3->link=h3;
            add_poly();
            display(h3);
          break;
       case 3:
          exit(0);
       default: printf("Invalid Choice\n");
       }

    }

}
```