```
/*
Assignment 4:
Devise a scheme in computing a polynomial in C where c is computed by:
a)  Adding 2 polynomials A and B
b)  Subtracting polynomial B from A
c)  Multiplying 2 polynomials A and B
d)  Differentiating polynomial A
*/

/*Including the header files*/
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

/*Declaring global variables*/
int poly[2][100],n1,m1,n2,m2,n3,m3;

int main()
{
    /*Declaring function prototypes and variables*/
    void input_poly();
    void add();
    void sub();
    void mult();
    void diff_a();
    void diff_b();
    void display1();
    void display2();
    int c,c1;
    /*Initializing index variables of the array*/
    n1=0;
    n2=0;
    n3=0;
    m1=0;
    m2=0;
    m3=0;
    /*Loop for user's choice to perform different operations on the entered polynomials*/
    do
    {
        printf("\n\tMENU");
        printf("\n1.Addition");
        printf("\n2.Subtraction");
        printf("\n3.Multiplication");
        printf("\n4.Differentiation of A");
        printf("\n5.Differentiation of B");
        printf("\n6.Exit");
        printf("\nEnter choice (1,2,3,4,5,6) :- ");
        scanf("%d",&c);
        switch(c)
        {
            /*Addition of the polynomials*/
            case 1:
                /*Checking if the polynomials are already entered and user wants to perform th
e operation on those data set itself*/
                if(m1 != 0 && m2 != 0)
                {
                    printf("\nDo you want to perform Addition on a new set of data elements or
 the existing one?(YES=1,NO=0) :-");
                    scanf("%d",&c1);
                    if(c1 == 1)
                        input_poly();
                }
                else
                    input_poly();
                add();
                display1();
                printf("\nAfter Addition the result is:-\n");
                display2();
                break;
            /*Subtraction of the polynomials*/
            case 2:
                /*Checking if the polynomials are already entered and user wants to perform th
e operation on those data set itself*/
                if(m1 != 0 && m2 != 0)
                {
                    printf("\nDo you want to perform Addition on a new set of data elements or
 the existing one?(YES=1,NO=0) :-");
                    scanf("%d",&c1);
                    if(c1 == 1)
                        input_poly();
                }
                else
```

```
                    input_poly();
                sub();
                display1();
                printf("\nAfter Subtraction the result is:-\n");
                display2();
                break;
            /*Multiplication of the polynomials*/
            case 3:
                /*Checking if the polynomials are already entered and user wants to perform th
e operation on those data set itself*/
                if(m1 != 0 && m2 != 0)
                {
                    printf("\nDo you want to perform Addition on a new set of data elements or
 the existing one?(YES=1,NO=0) :-");
                    scanf("%d",&c1);
                    if(c1 == 1)
                        input_poly();
                }
                else
                    input_poly();
                mult();
                display1();
                printf("\nAfter Multiplication the result is:-\n");
                display2();
                break;
            /*Differenting the 1st polynomial*/
            case 4:
                /*Checking if the polynomials are already entered and user wants to perform th
e operation on those data set itself*/
                if(m1 != 0 && m2 != 0)
                {
                    printf("\nDo you want to perform Addition on a new set of data elements or
 the existing one?(YES=1,NO=0) :-");
                    scanf("%d",&c1);
                    if(c1 == 1)
                        input_poly();
                }
                else
                    input_poly();
                diff_a();
                display1();
                printf("\nAfter Differentiating A the result is:-\n");
                display2();
                break;
            /*Differenting the 2nd polynomial*/
            case 5:
                /*Checking if the polynomials are already entered and user wants to perform th
e operation on those data set itself*/
                if(m1 != 0 && m2 != 0)
                {
                    printf("\nDo you want to perform Addition on a new set of data elements or
 the existing one?(YES=1,NO=0) :-");
                    scanf("%d",&c1);
                    if(c1 == 1)
                        input_poly();
                }
                else
                    input_poly();
                diff_b();
                display1();
                printf("\nAfter Differentiating B the result is:-\n");
                display2();
                break;
            case 6:
                exit(0);
            default:
                printf("\nWrong Input : Re-Enter");
                break;
        }
    }while(1);
    return(0);
}

/*Fuction that is used to enter the 2 polynomials*/
void input_poly()
{
    int i,c;
    /*Initializing the 1st polynomial index and entering the polynomial*/
    n1=0;
    m1=0;
    printf("\nEnter 1st Polynomial\n");
    i=0;
    do
    {
        printf("Enter Coeficient :- ");
        scanf("%d",&poly[0][m1]);
        printf("Enter Exponent :- ");
        scanf("%d",&poly[1][m1]);
        m1++;
        printf("Any more? (YES=1,NO=0) :- ");
        scanf("%d",&c);
    }while(c == 1);
    /*Initializing the 2nd polynomial index and entering the polynomial*/
    n2=m1;
    m2=m1;
    printf("\nEnter 2nd Polynomial\n");
    i=0;
    do
    {
        printf("Enter Coeficient :- ");
        scanf("%d",&poly[0][m2]);
        printf("Enter Exponent :- ");
        scanf("%d",&poly[1][m2]);
        m2++;
        printf("Any more elements? (YES=1,NO=0) :- ");
        scanf("%d",&c);
    }while(c == 1);
    n3=m2;
    m3=m2;
}

/*Function to display the 2 entered polynomials*/
void display1()
{
    int i;
    /*Displaying the 1st polynomial*/
    printf("\n1st Polynomial\n");
    i=n1;
    printf(" %dx^%d ",poly[0][n1],poly[1][n1]);
    for(i=n1+1;i<m1;i++)
    {
        if(poly[0][i] > 0)
            printf("+");
        printf(" %dx^%d ",poly[0][i],poly[1][i]);
    }
    /*Displaying the 2nd polynomial*/
    printf("\n2nd Polynomial\n");
    i=n2;
    printf(" %dx^%d ",poly[0][n2],poly[1][n2]);
    for(i=n2+1;i<m2;i++)
    {
        if(poly[0][i] > 0)
            printf("+");
        printf(" %dx^%d ",poly[0][i],poly[1][i]);
    }
}

/*Function to display the resultant polynomial*/
void display2()
{
    int i,j,k;
    /*Compressing the resultant polynomial*/
    i=n3;
    for(i=n3;i<m3;i++)
    {
        for(j=i+1;j<m3;j++)
        {
            if(poly[1][i] == poly[1][j])
            {
                poly[0][i]=poly[0][i]+poly[0][j];
                for(k=j;k<m3;k++)
                {
                    poly[0][k]=poly[0][k+1];
                    poly[1][k]=poly[1][k+1];
                }
                m3--;
            }
        }
    }
    /*Displaying the resultant polynomial*/
    i=n3;
    printf(" %dx^%d ",poly[0][n3],poly[1][n3]);
    for(i=n3+1;i<m3;i++)
    {
        if(poly[0][i] > 0)
            printf("+");
        printf(" %dx^%d ",poly[0][i],poly[1][i]);
    }
}

/*Function to add the entered polynomials*/
void add()
{
    int i,j,f;
    /*Initializing the resultant polynomial index*/
    n3=m2;
    m3=m2;
    /*Performing the addition operation with respect to the 1st polynomial*/
    for(i=n1;i<m1;i++)
    {
        f=0;
        for(j=n2;j<m2;j++)
        {
            if(poly[1][i] == poly[1][j])
            {
                poly[0][m3]=poly[0][i]+poly[0][j];
                poly[1][m3]=poly[1][i];
                m3++;
                f=1;
                break;
            }
        }
        if(f == 0)
        {
            poly[0][m3]=poly[0][i];
            poly[1][m3]=poly[1][i];
            m3++;
        }
    }
    /*Entering the 2nd polynomial terms that have not yet been added*/
    for(i=n2;i<m2;i++)
    {
        f=0;
        for(j=n3;j<m3;j++)
        {
            if(poly[1][i] == poly[1][j])
            {
                f=1;
                break;
            }
        }
        if(f == 0)
        {
            poly[0][m3]=poly[0][i];
            poly[1][m3]=poly[1][i];
            m3++;
        }
    }
}

/*Function to subtract the entered polynomials*/
void sub()
{
    int i,j,f;
    /*Initializing the resultant polynomial index*/
    n3=m2;
    m3=m2;
    /*Performing the subtraction operation with respect to the 1st polynomial*/
    for(i=n1;i<m1;i++)
    {
        f=0;
        for(j=n2;j<m2;j++)
        {
            if(poly[1][i] == poly[1][j])
            {
                poly[0][m3]=poly[0][i]-poly[0][j];
                poly[1][m3]=poly[1][i];
                m3++;
                f=1;
                break;
            }
        }
        if(f == 0)
        {
            poly[0][m3]=poly[0][i];
            poly[1][m3]=poly[1][i];
            m3++;
        }
    }
    /*Entering the 2nd polynomial terms that have not yet been subtracted*/
    for(i=n2;i<m2;i++)
    {
        f=0;
        for(j=n3;j<m3;j++)
        {
            if(poly[1][i] == poly[1][j])
            {
                f=1;
                break;
            }
        }
        if(f == 0)
        {
            poly[0][m3]=-1*poly[0][i];
            poly[1][m3]=poly[1][i];
            m3++;
        }
    }
}

/*Function to multiply the entered polynomials*/
void mult()
{
    int i,j;
    /*Initializing the resultant polynomial index*/
    n3=m2;
    m3=m2;
    /*Performing the multiplication operation on the polynomials*/
    for(i=n1;i<m1;i++)
    {
        for(j=n2;j<m2;j++)
        {
            poly[0][m3]=poly[0][i]*poly[0][j];
            poly[1][m3]=poly[1][i]+poly[1][j];
            m3++;
        }
    }
}

/*Function to differentiate the 1st polynomial*/
void diff_a()
{
    int i;
    /*Initializing the resultant polynomial index*/
    n3=m2;
    m3=m2;
    for(i=n1;i<m1;i++)
    {
        poly[0][m3]=poly[0][i]*poly[1][i];
        poly[1][m3]=poly[1][i]-1;
        m3++;
    }
}

/*Function to differentiate the 2nd polynomial*/
void diff_b()
{
    int i;
    /*Initializing the resultant polynomial index*/
    n3=m2;
    m3=m2;
    for(i=n2;i<m2;i++)
    {
        poly[0][m3]=poly[0][i]*poly[1][i];
        poly[1][m3]=poly[1][i]-1;
        m3++;
    }
}
```