CISCO- Advanced C Project

****

Disclaimer:

This Document Contains the Project Approach, Problem Statement, and Solution Submitted by Me for CISCO NCH Advanced C Project.

Email: ***devangsharma419@gmail.com***

****

SOURCE CODE:

```
/* Solution Submitted By: DEVANG SHARMA
Email: devangsharma419@gmail.com
Alternate Email: devangsharma25398@gmail.com
T: +91-9953027469
*/


// Code Starts Here

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

void solvebysubstitution(float a,float b,float c,float d,float
e,float f)
```

```
{
        // Vars x and y used to store answer
    float x,y;

    float det; // Used to hold 'determinent' value

    float coeff[2][2]; // Multi-Dimensional array to hold
coefficients
    float values[2];

    coeff[0][0] = a;
    coeff[0][1] = b;
    coeff[1][0] = d;
    coeff[1][1] = e;

    values[0] = c;
    values[1] = f;

    det = ((coeff[0][0] * coeff[1][1]) -
(coeff[0][1]*coeff[1][0]));

    printf("\n\n ***Solution By Substitution Method*** \n");

    if (det == 0)
    {
        printf("\n *Solution Does Not Exist*\n");
        exit(1);
    }

    else
    {
        x = (((coeff[1][1] * values[0]) - (coeff[0][1] *
values[1])) / det);
```

```
        y = (((coeff[0][0] * values[1]) - (coeff[1][0] *
values[0])) / det);
        printf("\n X = %f", x);
        printf("\n Y = %f\n", y);
    }
}



void solvebymatrix (float a, float b, float c, float d, float e,
float f)
{
    int n,i,j,k;
    n=2;

    float mat[2][3],x[2];

    mat[0][0]=a;
    mat[0][1]=b;
    mat[0][2]=c;
    mat[1][0]=d;
    mat[1][1]=e;
    mat[1][2]=f;

    for (i=0;i<n;i++)                    //Pivotisation
        for (k=i+1;k<n;k++)
            if (abs(mat[i][i]) < abs(mat[k][i]))
                for (j=0;j<=n;j++)
                {
                    double temp=mat[i][j];
                    mat[i][j]=mat[k][j];
                    mat[k][j]=temp;
                }
```

```
    printf("\n ***Solution By Matrix Method*** \n");

    printf("\nThe matrix after Pivotisation is:\n");
    for (i=0;i<n;i++)            //print the new matrix
    {
        for (j=0;j<=n;j++)
            printf("%f ",mat[i][j]);
        printf("\n");
    }

    for (i=0;i<n-1;i++)            //loop to perform the gauss
elimination
        for (k=i+1;k<n;k++)
            {
                double t=mat[k][i]/mat[i][i];
                for (j=0;j<=n;j++)
                    mat[k][j]=mat[k][j]-t*mat[i][j];    //make
the elements below the pivot elements equal to zero or elimnate
the variables
            }

    printf("\nThe matrix after gauss-elimination is as
follows:\n");
    for (i=0;i<n;i++)            //print the new matrix
    {
        for (j=0;j<=n;j++)
            printf("%f ",mat[i][j]);
        printf("\n");
    }

    for (i=n-1;i>=0;i--)
    {
```

```
        x[i]=mat[i][n];
        for (j=i+1;j<n;j++)
            if (j!=i)
                x[i]=x[i]-mat[i][j]*x[j];
        x[i]=x[i]/mat[i][i];
    }

    printf("\nThe values of the variables are as follows:\n");
        printf("X is: %f\n", x[0]);
        printf("Y is: %f", x[1]);

 }

// ax+by=c
int func(int x, int a, int b, int c)
{

    int y=(c-a*x)/b;
    return y;
}

// dx+ey=f
int func2(int x, int d, int e, int f)
{
    int y=(f-d*x)/e;
    return y;
}

int main(void)
{
    // Vars a-f used to hold user input
    float a,b,c,d,e,f;
```

```
    printf("Provide the value of a, b and c: ");
    scanf("%f %f %f", &a, &b, &c);

    printf("\nProvide the value of d, e and f: ");
    scanf("%f %f %f", &d, &e, &f);

    solvebysubstitution(a,b,c,d,e,f);
    solvebymatrix(a,b,c,d,e,f);

    printf("\n ****Printing the Graph:**** \n");
    int i=0,j=0;

    for (i=-10;i<10;i++)
    {
        printf("\n");
        for (j=-10;j<10;j++)
        {
            if (func(i, (int)a, (int)b, (int)c)==j)
            printf("2");

            else if (func2(i, (int)d, (int)e ,(int)f)==j)
            printf("1");

            else
            printf(" ");
        }
    }


    return 0;

}
```