

# CISCO- Advanced C Project

\*\*\*\*\*

## Disclaimer:

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

All Copyrights Reserved By: Devang Sharma

Email: devangsharma419@gmail.com

\*\*\*\*\*

## Problem Statement

### DESCRIPTION

Simultaneous equations are those equations involving two or more unknowns that are to have the same values in each equation. They occur as a pair of equations, usually represented by unknowns such as x and y. A sample set of simultaneous set of equations go like the following:

$$3x+2y=7$$

$$5x-3y=37$$

### Part 1

Your task is to develop a C program that will solve two simultaneous linear equations of the above forms using 2 methods:

1. Substitution method
2. Matrix method.

In this part, you should solve the equation by both the methods and compare the solution.

Your application should accept the parameters, from the console, and construct the equations. For example, it should ask you to prompt for 'a', 'b' and 'c' and 'p', 'q' and 'r' to construct the equations.

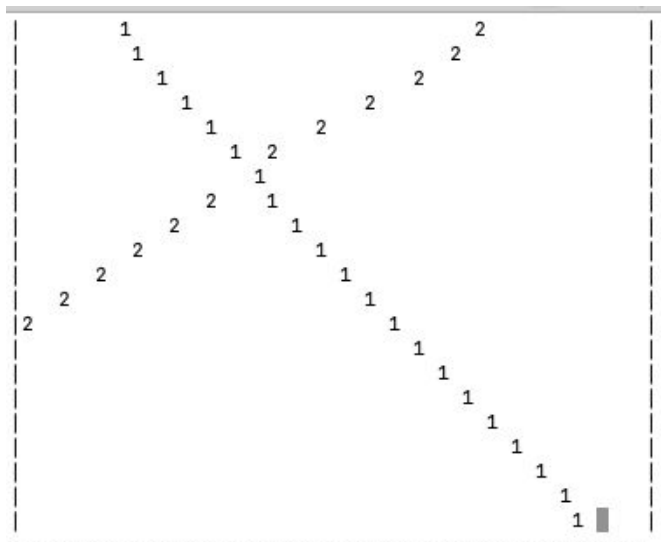
$ax+by=c$ , and

$px+qy=r$

When performing the computation using the matrix method, your application should indicate the intermediary results such as determinant, inverted matrix etc.

As you are aware, for any two given linear equations, when there is a solution, these two lines intersect. When there is no solution, these two lines don't intersect, and, could be that they run parallel, for example.

## Part 2



When there is a solution, ie, when these two lines are intersecting, you should plot the lines using the characters on the console in the following format: This is a sample graph, not to scale and is not representing the given set of equations.

Your application should be able to demonstrate that it can handle any set of equations, and when there is a solution, it can plot a graph on the console, as above, else it will be able to let you know that there is no solution.

## Assumptions

Students are aware of basic algebra and know the ways solutions are computed. Students are aware of using c, libraries, functions etc. to create the project.

### **Sample Data**

The linear equations set having a solution:

$$3x + 2y - 7 = 0, \text{ and}$$

$$5x - 3y - 37 = 0$$

The linear equations set having no solution (parallel lines):

$$3x - y + 2 = 0$$

$$-3x + y = 0$$

### **Note:**

The problem can be treated programmatically, by adopting 2 different solution methods: substitution method and matrix manipulation method (using matrix inversion, matrix multiplication and other functions).

In part 2, the graph is to be plotted.

# Solution

## By Substitution Method:

```
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);
    }
}

```

## **By Matrix Method**

```

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 eliminate
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]);

}

```

## Plotting the Graph

```
// 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;
}
```

## Inside The Main:

```
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(" ");  
    }  
}
```