**Problem No: 01**

**Problem Name:** Check whether the system is consistent and Find the solution if it is consistent

4x1 + 3x2 – x3 = 6

3x1 + 5x2 + 3x3 = 4

x1 + x2 + x3 = 1

**Objective:** To learn consistency of a system and to find the solution of a system.

**Source Code:**

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

double det(int n, double mat[4][4]);

int rankA, rankab;

double x, y, z;

void printMat(int r, int c, double matA[4][4]);

void solAB(int r, int c, double matAB[4][4]);

int main()

{

int flag = 0;

double matA[4][4] = {{4, 3, -1}, {3, 5, 3}, {1, 1, 1}};

double subA[4][4] = {{4, 3}, {3, 5}};

double matAB[4][4] = {{4, 3, -1, 6}, {3, 5, 3, 4}, {1, 1, 1, 1}};

double matB[4][4] = {{6}, {4}, {1}};

if(det(3, matA) != 0) rankA = 3;

else if(det(2, subA) != 0) rankA = 2;

else rankA = 1;

cout << "Matrix A: \n";

printMat(3, 3, matA);

cout << "Matrix B: \n";

printMat(3, 1, matB);

cout << "Matrix AB: \n";

printMat(3, 4, matAB);

cout << "Determinant of Matrix A: " << det(3, matA) << endl;

if(det(3, matA) == 0){

cout << "Sub-Matrix of A: \n";

printMat(2, 2, subA);

cout << "Determinant of Sub-Matrix A: " << det(2, subA) << endl;

}

cout << "Rank of Matrix A: " << rankA << endl;

cout << "\nEcsholon form of Matrix AB:\n";

solAB(3, 4, matAB);

cout << "Rank of Matrix AB: " << rankab << endl;

if(rankA == rankab){

flag = 1;

cout << "\nSystem is consistent. Solution exists.\n";

}

else

cout << "\nSystem is not consistent. Solution doesn't exists.\n";

if(flag = 1){

cout << "\nSolution is:\n";

cout << "x: " << x << endl;

cout << "y: " << y << endl;

cout << "z: " << z << endl;

}

else

return 0;

return 0;

}

void printMat(int r, int c, double matA[4][4])

{

for(int i = 0; i < r; i++){

for(int j = 0; j < c; j++){

cout << fixed << setprecision(2) << matA[i][j] << " ";

}

}

}

double det(int n, double mat[4][4])

{

int c, subi, i, j, subj;

double d;

double submat[4][4];

if (n == 2) {

return( (mat[0][0] \* mat[1][1]) - (mat[1][0] \* mat[0][1]));

}

else{

for(c = 0; c < n; c++){

subi = 0;

for(i = 1; i < n; i++){

subj = 0;

for(j = 0; j < n; j++){

if (j == c)

continue;

submat[subi][subj] = mat[i][j];

subj++;

}

subi++;

}

d += (pow(-1 ,c) \* mat[0][c] \* det(n - 1 ,submat));

}

}

return d;

}

void solAB(int r, int c, double matAB[4][4])

{

int i, j;

for(j = 0; j < c; j++){

matAB[0][j] /= 4;

}

printMat(3, 4, matAB);

for(j = 0; j < c; j++)

matAB[1][j] -= (matAB[0][j] \* 3);

printMat(3, 4, matAB);

for(j = 0; j < c; j++)

matAB[2][j] -= matAB[0][j];

printMat(3, 4, matAB);

for(j = 0; j < c; j++)

matAB[2][j] -= (matAB[1][j] / 11);

printMat(3, 4, matAB);

if(matAB[2][2] != 0) rankab = 3;

else if(matAB[2][2] == 0 && matAB[2][3] == 0 && matAB[1][1] != 0) rankab = 2;

z = matAB[2][3] / matAB[2][2];

y = (matAB[1][3] - z \* matAB[1][2]) / matAB[1][1];

x = (matAB[0][3] - (z \* matAB[0][2]) - (y \* matAB[0][1])) / matAB[0][0];

}

**Output:**

