**PRACTICAL NO – 3(B)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Name :

Roll No :

Aim : Program for solving linear system of equations using Gauss -Jordon methods.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function [**x**]=gauss\_jordon(**A**, **B**)

[nA,mA]=size(**A**);

[nb,mb]=size(**B**);

d=det(**A**);

if d==0 then

printf('Matrix is non singular solution does not exist');

abort;

end;

A\_aug =[**A** eye(nA,nA)];

A\_aug\_inv = rref(A\_aug);

A\_inv\_1=[A\_aug\_inv(:,4),A\_aug\_inv(:,5),A\_aug\_inv(:,6)];

**x**=A\_inv\_1\***B**;

printf('x=%g\ty==%g\tx=%g\n',**x**(1,1),**x**(2,1),**x**(3,1));

return(**x**);

endfunction

**OUTPUT :**

-1->A=[5,1,1;1,7,1;1,1,6]

A =

5. 1. 1.

1. 7. 1.

1. 1. 6.

-1->B=[4;12;-5]

B =

4.

12.

- 5.

-1->gauss\_jordon(A,B)

x=0.690722 y==1.79381 x=-1.24742

ans =0.6907216

1.7938144

- 1.2474227