```
clear all;
close all;

A = [1e-10 4;2 1];

b = [1;1];

%Exact solution
fprintf('True Solution;\n');
U_exact = A\b

%LU without partial pivoting
[L, U] = lu_wopp(A);
fprintf('Solution Obtained without partial pivoting;\n');
x_opp = lu_solve1(L,U,b)

%LU with partial pivoting
[L,U,P,pv] = lu_bug_pp(A);
fprintf('Solution Obtained with partial pivoting;\n');
x_wpp = lu_solve(L,U,b,pv)

%Error of LU with partial pivoting
fprintf('Error Obtained with partial pivoting;\n');
error_wpp = abs(U_exact - x_wpp)
fprintf('Error Obtained without partial pivoting;\n');
error_wopp = abs(U_exact - x_opp)

fprintf('Using partial pivoting we obatin exact values because we obtain zero error,\n while without partial pivoting we obtained a slightly smal

fprintf('d). While doing LU decomposition, we need to create an upper triangular matrix U, by making\n  entry a_21 = 0 in matrix A. We shall have
```

```
True Solution;

U_exact =

    0.3750
    0.2500

Solution Obtained without partial pivoting;

x_opp =

    0.3750
    0.2500

Solution Obtained with partial pivoting;

x_wpp =

    0.3750
    0.2500

Error Obtained with partial pivoting;

error_wpp =

     0
     0

Error Obtained without partial pivoting;

error_wopp =

   1.0e-07 *

    0.3102
         0

Using partial pivoting we obatin exact values because we obtain zero error,
 while without partial pivoting we obtained a slightly smaller error
d). While doing LU decomposition, we need to create an upper triangular matrix U, by making
  entry a_21 = 0 in matrix A. We shall have to perform a calculation R2 <-- (1e-10)R2  - 2R1,
 which will become (1e-10)(1) -2(4), so we shall have a very small number in magnitude minus
 a big number in magnitude: 8. Normally this must give us -8, but due to catastrophic loss
 of accuracy we obtain -7.9999999998 hence catastrophic cancellation.
```

Published with MATLAB® R2020a