CS 3320 – Numerical Software
Module 8 – Program
"LU Factorization"

Write a program, named `lud.py`, that solves linear systems using the LU factorization with partial pivoting. Separate the factorization and the solution into two functions:

1. `factor(A, n, pivot)`
    a. `A` is the square matrix of coefficients
    b. `n` is the rank (number of rows/columns) of `A`
    c. `pivot` is an output vector that records the partial pivoting swaps
2. `solve(A, n, pivot, b, x)`
    a. `A`, `n` and `pivot` are from `factor()` (`A` is overwritten with its own LU factorization)
    b. `b` is a right-hand side to solve for
    c. `x` is an output vector with the solution of `Ax = b`

Write your `main()` function to process an arbitrary number of input files. Each file contains information for one linear system, with possibly multiple right-hand sides to solve for. For example, if the files lu1.dat and lu2.dat are as follows:

File lu1.dat:
```
3
10 -7 0
-3 2 6
5 -1 5
2
7 4 6
3 5 9
```

File lu2.dat:
```
4
5 3 -1 0
2 0 4 1
-3 3 -3 5
0 6 -2 3
2
11 1 -2 9
7 7 2 7
```

then the expected output to 2 decimals is as follows:

```
System Prompt\> prog6 lu1.dat lu2.dat
File lu1.dat:
L\U = 10.00    -7.00     0.00
       0.50          2.50      5.00
      -0.30     -0.04     6.20
b =  7.00 4.00 6.00
x =  0.00 -1.00 1.00
b =  3.00 5.00 9.00
x =  1.00 1.00 1.00


File lu2.dat:
```

```
L\U =5.00 3.00 -1.00 0.00
     0.00 6.00 -2.00 3.00
     0.40 -0.20 4.00 1.60
     -0.60 0.80 -0.50 3.40
b =  11.00 1.00 -2.00 9.00
x =  1.00 2.00 -0.00 -1.00
b =  7.00 7.00 2.00 7.00
x =  1.00 1.00 1.00 1.00
```

The layout of the input files is:

```
<rank of system (number of rows)>
<the rows of the matrix, a>
<number of right-hand sides>
<the right hand sides>
```

Note that you are printing out the LU factorization that overwrites `A`. Be sure to check for invalid input, such as missing data (if you run out of numbers, print a message and continue to the next file). If you cannot find a non-zero pivot element during partial pivoting in `factor()`, then the system either has no solution or infinite number of solutions, return an error status (-1). Your program will then print a message indicating that the A matrix is singular then continue to the next file.