Assignment 6

```
A=[1 -5 -4 -9 5 ; -3 0 4 -1 0 ; -8 -3 -9 -3 -2 ; 8 -8 -6 9 5 ; 4 7 -4 0 -3]
```

```
A = 5×5
     1    -5    -4    -9     5
    -3     0     4    -1     0
    -8    -3    -9    -3    -2
     8    -8    -6     9     5
     4     7    -4     0    -3
```

```
b=[-9 ; -4 ; 7 ; 6 ; -2]
```

```
b = 5×1
    -9
    -4
     7
     6
    -2
```

```
%from lab 5
function [L,U]=LUfact(A) % This function gives A=LU when row exchanges are
not
%required.
if size(A,1)~=size(A,2) %this means if #rows is not equal to #columns
disp('LU factorization is possible only for square matrices')
else %so now A is square
n=size(A,1); % n is the size of the square matrix
U=A; % Initialize U as A and perform row operations
% until it becomes upper triangular.
L=eye(n); % L is initially the identity matrix
for j=1:n-1 %j refers to the columns. To create an upper
% triangular form we eliminate all columns apart from the last one
for i=j+1:n %i refers to the rows. We eliminate below the main diagonal
% so i is greater than j ( which is our column right now)
if U(j,j)~=0 % we will eliminate using the coefficient U(i,j)/U(j,j)
%If that coefficient was zero then we can't use it and move on
L(i,j)= U(i,j)/U(j,j); % the coefficient that we used to eliminate

%the (i,j) entry of U becomes the (i,j) entry of L
U(i,1:n)=U(i,1:n) - (U(i,j)/U(j,j))*U(j,1:n); % we eliminate the entry
% (i,j) by subtracting U(i,j)/U(j,j) times the jth row of U from
% the ith row
end
end
end
L % After evereything is over we show what the matrices L,U are
U
end
end

LUfact(A)
```

```
L = 5×5
    1.0000         0         0         0         0
   -3.0000    1.0000         0         0         0
   -8.0000    2.8667    1.0000         0         0
    8.0000   -2.1333   -0.4945    1.0000         0
    4.0000   -1.8000    0.1328   -0.6326    1.0000
U = 5×5
    1.0000   -5.0000   -4.0000   -9.0000    5.0000
         0  -15.0000   -8.0000  -28.0000   15.0000
         0         0  -18.0667    5.2667   -5.0000
         0         0         0   23.8708   -5.4723
         0         0         0    0.0000    1.2027
ans = 5×5
    1.0000         0         0         0         0
   -3.0000    1.0000         0         0         0
   -8.0000    2.8667    1.0000         0         0
    8.0000   -2.1333   -0.4945    1.0000         0
    4.0000   -1.8000    0.1328   -0.6326    1.0000
```

```
function x = backward(A,b)
% square matrix with non-zero diagonals
    n=size(A,1);
    x = zeros(n,1);
    x(n) = b(n)/A(n,n);
    for i = n-1:-1:1
        x(i)=(b(i)-A(i,i+1:n)*x(i+1:n))/A(i,i);
    end
end
```

```
backward(A,b)
```

```
ans = 5×1
       I
       I
   -1.02
    0.29
    0.66
```

```
function x = forward(A,b)
% triangular square matrix with non-zero diagonals
    n=size(A,1);
    x = zeros(n,1);
    x(1) = b(1)/A(1,1);
    for i = 2:n
        x(i)=(b(i)-A(i,1:i-1)*x(1:i-1))/A(i,i);
    end
end
```

```
forward(A,b)
```

```
ans = 5×1
    -9
```

```
  -Inf
   Inf
   NaN
   NaN
```

```
function x=mySolve(A,b)
[L,U]=LUfact(A);
y=forward(L,b); %where y=U*x
x=backward(U,y);
end
```

```
mySolve(A,b)
```

```
L = 5×5
    1.0000         0         0         0         0
   -3.0000    1.0000         0         0         0
   -8.0000    2.8667    1.0000         0         0
    8.0000   -2.1333   -0.4945    1.0000         0
    4.0000   -1.8000    0.1328   -0.6326    1.0000
U = 5×5
    1.0000   -5.0000   -4.0000   -9.0000    5.0000
         0  -15.0000   -8.0000  -28.0000   15.0000
         0         0  -18.0667    5.2667   -5.0000
         0         0         0   23.8708   -5.4723
         0         0         0    0.0000    1.2027
ans = 5×1
    2.5893
   -4.9210
    0.7134
   -0.9145
   -8.3143
```