Assignment 2

Ex4.23

```
M = [
    4, -1, 3, 2;
    -8, 0 ,-3, -3.5;
    2, -3.5, 10, 3.75;
    -8, -4, 1, -0.5
];

[L,U] = LUdecompGauss(M);
A = L*U;

disp(M);
disp(L);
disp(U);
disp(A);

function [L, U] = LUdecompGauss(A)
% Decomposes A into lower triangular and upper triangular
% matrices (L and U) using Gauss elemination, without
% pivoting, such that [A] = [L][U]

    [m,n] = size(A);
    if m ~= n
        disp('The matrix must be square.');
        L = 0;
        U = 0;
        return
    end

    L = eye(n);     % start with Identity matrix for L
    U = A;          % copy of A for U
    for i = 1:n-1       % for every row (but the last)...
        for j = i+1:n   % do the following for all remaining rows:
            L(j,i) = U(j,i)/U(i,i); % Calculate L(j,i) ( U(i,i) is the pivot coefficient )
            for k = 1:n
                U(j,k) = U(j,k) - (L(j,i) * U(i,k));  % subtract the pivot equation times
L(i,j)
            end                                     % from U(j,k) -> U(i,j) becomes zero
        end
    end
end
```

Ex5.17

a)

```
A = [
    0, 0, 0, 1, 0, 0;
    1, 0, 1, 0, 1, 1;
    0, 1, 0, 0, 1, 0;
    1, 1, 0, 0, 1, 0;
    1, 1, 1, 0, 0, 1;
    1, 0, 0, 0, 1, 0
];

[V, D] = eig(A);
disp(V);
disp(D);
```

Eigen Vectors (V):

```
0.1761 0.3379 0.0000 -0.5773 -0.5773 0.5774
0.5155 -0.1443 0.0000 -0.0000 -0.0000 -0.0000
0.3938 -0.7555 -0.7071 0.0000 0.0000 0.0000
0.4611 0.1290 0.0000 0.5774 0.5774 -0.5773
0.5155 -0.1443 -0.0000 -0.0000 -0.0000 -0.0000
0.2642 0.5068 0.7071 0.5773 0.5773 -0.5774
```

Eigen Values (diagonal of D): 2.618, 0.382, 0, -1, -1, -1

b)

Eigen Vector -> 0.1761, 0.5155, 0.3938, 0.4611, 0.5155, 0.2642

Rank	Team	Eigen Vector Value
1st	2	0.5155
1st	5	0.5155
3rd	4	0.4611
4th	3	0.3938
5th	6	0.2642
6th	1	0.1761