```
% Question 01:

A = randi(10,2,2) % Generate a random 2x2 matrix
```

```
A = 2×2
     4     7
     1     9
```

```
B = A;              % Initialize B to be equal to A
for i = 1:10        % Repeat the process 10 times
    [Q,R] = qr(B,0) % Decompose B into the product of an orthogonal matrix Q and an upp
    B = R*Q         % Update B to be equal to R*Q
end
```

```
Q = 2×2
   -0.9701   -0.2425
   -0.2425    0.9701
R = 2×2
   -4.1231   -8.9738
         0    7.0335
B = 2×2
    6.1765   -7.7059
   -1.7059    6.8235
Q = 2×2
   -0.9639    0.2662
    0.2662    0.9639
R = 2×2
   -6.4077    9.2444
         0    4.5258
B = 2×2
    8.6375    7.2049
    1.2049    4.3625
Q = 2×2
   -0.9904   -0.1382
   -0.1382    0.9904
R = 2×2
   -8.7212   -7.7385
         0    3.3252
B = 2×2
    9.7066   -6.4594
   -0.4594    3.2934
Q = 2×2
   -0.9989    0.0473
    0.0473    0.9989
R = 2×2
   -9.7175    6.6079
         0    2.9843
B = 2×2
   10.0190    6.1411
    0.1411    2.9810
Q = 2×2
   -0.9999   -0.0141
   -0.0141    0.9999
R = 2×2
  -10.0200   -6.1824
         0    2.8942
B = 2×2
   10.1061   -6.0408
   -0.0408    2.8939
Q = 2×2
```

```
   -1.0000    0.0040
    0.0040    1.0000
R = 2×2
  -10.1062    6.0524
         0    2.8695
B = 2×2
   10.1305    6.0116
    0.0116    2.8695
Q = 2×2
   -1.0000   -0.0011
   -0.0011    1.0000
R = 2×2
  -10.1305   -6.0148
         0    2.8626
B = 2×2
   10.1374   -6.0033
   -0.0033    2.8626
Q = 2×2
   -1.0000    0.0003
    0.0003    1.0000
R = 2×2
  -10.1374    6.0042
         0    2.8607
B = 2×2
   10.1393    6.0009
    0.0009    2.8607
Q = 2×2
   -1.0000   -0.0001
   -0.0001    1.0000
R = 2×2
  -10.1393   -6.0012
         0    2.8602
B = 2×2
   10.1398   -6.0003
   -0.0003    2.8602
Q = 2×2
   -1.0000    0.0000
    0.0000    1.0000
R = 2×2
  -10.1398    6.0003
         0    2.8600
B = 2×2
   10.1400    6.0001
    0.0001    2.8600
```

```
eigenvalues = diag(B) % Extract the eigenvalues from the diagonal of B
```

```
eigenvalues = 2×1
   10.1400
    2.8600
```

```
% Question 02:

% Part 01-
A = [9 10; 10 7; 2 1] % Define the matrix A
```

```
A = 3×2
     9    10
    10     7
     2     1
```

```
[U,Z,V] = svd(A)       % Compute the SVD of A
```

```
U = 3×3
   -0.7359    0.6692    0.1031
   -0.6666   -0.6894   -0.2835
   -0.1186   -0.2773    0.9534
Z = 3×2
   18.1781         0
         0    2.1348
         0         0
V = 2×2
   -0.7441   -0.6681
   -0.6681    0.7441
```

```
Z = svd(A)            % Extract the singular values of A
```

```
Z = 2×1
   18.1781
    2.1348
```

```
singular_values = diag(Z) % Extract the singular values from the diagonal matrix Z
```

```
singular_values = 2×2
   18.1781         0
         0    2.1348
```

```
% Part 02-

U_orthogonal = U'*U   % Verify that U is orthogonal
```

```
U_orthogonal = 3×3
    1.0000    0.0000    0.0000
    0.0000    1.0000   -0.0000
    0.0000   -0.0000    1.0000
```

```
U_orthogonal = U*U'   % Verify that U is orthogonal
```

```
U_orthogonal = 3×3
    1.0000    0.0000    0.0000
    0.0000    1.0000   -0.0000
    0.0000   -0.0000    1.0000
```

```
V_orthogonal = V'*V   % Verify that V is orthogonal
```

```
V_orthogonal = 2×2
     1     0
     0     1
```

```
V_orthogonal = V*V'   % Verify that V is orthogonal
```

```
V_orthogonal = 2×2
     1     0
     0     1
```

```
% Part 03-
```

```
A = [9 10; 10 7; 2 1] % Define the matrix A
```

```
A = 3×2
     9    10
    10     7
     2     1
```

```
[U,Z,V] = svd(A)      % Compute the SVD of A
```

```
U = 3×3
   -0.7359    0.6692    0.1031
   -0.6666   -0.6894   -0.2835
   -0.1186   -0.2773    0.9534
Z = 3×2
   18.1781         0
         0    2.1348
         0         0
V = 2×2
   -0.7441   -0.6681
   -0.6681    0.7441
```

```
[V1,L1] = eig(A'*A)   % Compute the eigenvectors and eigenvalues of A'*A
```

```
V1 = 2×2
    0.6681   -0.7441
   -0.7441   -0.6681
L1 = 2×2
    4.5575         0
         0  330.4425
```

```
V                     % Print the matrix V
```

```
V = 2×2
   -0.7441   -0.6681
   -0.6681    0.7441
```

```
V1                    % Print the matrix V1
```

```
V1 = 2×2
    0.6681   -0.7441
   -0.7441   -0.6681
```

```
% Part 04-
[U1, L] = eig(A*A')
```

```
U1 = 3×3
   -0.1031    0.6692    0.7359
    0.2835   -0.6894    0.6666
   -0.9534   -0.2773    0.1186
L = 3×3
    0.0000         0         0
         0    4.5575         0
         0         0  330.4425
```

```
U                     % Print the matrix U
```

```
U = 3×3
   -0.7359    0.6692    0.1031
   -0.6666   -0.6894   -0.2835
   -0.1186   -0.2773    0.9534
```

```
% Part 05-
eig(A'*A)
```

```
ans = 2×1
    4.5575
  330.4425
```

```
eig(A*A')
```

```
ans = 3×1
    0.0000
    4.5575
  330.4425
```

```
S = diag(Z)         % singular values of A are diagonal elements of Z
```

```
S = 2×1
   18.1781
    2.1348
```

```
Ssquare = S.^2      % square of singular values
```

```
Ssquare = 2×1
  330.4425
    4.5575
```

```
% Question 04: Find Rank & Nullity
% Part 01-
A = [9 10; 10 7; 2 1]
```

```
A = 3×2
     9    10
    10     7
     2     1
```

```
r = rank(A);
m = size(A);
nullity = m(1)-r
```

```
nullity = 1
```

```
% Part 02-
% Verify that the third column in U forms a basis for the left nullspace of A.
A'*U(:,3)
```

```
ans = 2×1
1.0e-15 *
    0.1110
   -0.4441
```

```
% Part 03-
A = [1 1 0; 0 1 1];
[U,Z,V] = svd(A)
```

```
U = 2×2
   -0.7071    0.7071
   -0.7071   -0.7071
Z = 2×3
    1.7321         0         0
         0    1.0000         0
V = 3×3
   -0.4082    0.7071    0.5774
   -0.8165   -0.0000   -0.5774
   -0.4082   -0.7071    0.5774
```

```
pseudoA = V*pinv(Z)*U'
```

```
pseudoA = 3×2
    0.6667   -0.3333
    0.3333    0.3333
   -0.3333    0.6667
```

```
verification = pinv(A)     % Verify if this is same as pseudoA
```

```
verification = 3×2
    0.6667   -0.3333
    0.3333    0.3333
   -0.3333    0.6667
```

```
% ------------------------------------------------------------------------

% Exercise Question 01:

% Generate a random integer symmetric matrix of order 15
A = randi([-10, 10], 15, 15);
A = (A + A') / 2;

% Perform QR decomposition
[Q, R] = qr(A);

% Iterate until the matrix becomes diagonal
for i = 1:1000
    [Q, R] = qr(R);
end

% The diagonal entries of R are the eigenvalues of A
eigenvalues = diag(R);
disp(eigenvalues);
```

```
  -14.8240
   12.9129
   18.7002
  -11.3669
   17.5708
    9.5304
   10.6405
  -10.7539
  -14.8055
   -5.6843
  -14.1012
```

```
  -11.1061
    4.8214
   -4.6535
  -12.3752
```

```
[eigen_vec,eigen_val]=eig(A);    % Verification
disp(eigen_val);
```

```
  Columns 1 through 13

  -27.7958         0         0         0         0         0         0         0         0         0
         0  -21.9921         0         0         0         0         0         0         0         0
         0         0  -17.2533         0         0         0         0         0         0         0
         0         0         0  -16.3283         0         0         0         0         0         0
         0         0         0         0  -12.1450         0         0         0         0         0
         0         0         0         0         0   -6.7910         0         0         0         0
         0         0         0         0         0         0   -4.6370         0         0         0
         0         0         0         0         0         0         0    2.1782         0         0
         0         0         0         0         0         0         0         0    3.0599         0
         0         0         0         0         0         0         0         0         0    5.4921
         0         0         0         0         0         0         0         0         0         0    8.
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0

  Columns 14 through 15

         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
   21.4781         0
         0   28.6131
```

```
%--------------------------------------------------------------------

% Exercise Q-02:
% (a)
A = rand(7,9);
A(:,7:9) = A(:,1:3);
rankA = sprank(A);
disp('Matrix A: ');
```

```
Matrix A:
```

```
disp(A);
```

```
    0.4550    0.4366    0.8572    0.3476    0.4333    0.8030    0.4550    0.4366    0.8572
    0.1273    0.0492    0.9636    0.4612    0.8842    0.9995    0.1273    0.0492    0.9636
    0.0086    0.0496    0.4889    0.6393    0.3931    0.9810    0.0086    0.0496    0.4889
    0.7271    0.0911    0.2203    0.9173    0.1790    0.1270    0.7271    0.0911    0.2203
    0.3541    0.5940    0.2262    0.1616    0.6333    0.2322    0.3541    0.5940    0.2262
    0.7804    0.2411    0.5368    0.7156    0.6240    0.0236    0.7804    0.2411    0.5368
    0.4367    0.8414    0.7621    0.5777    0.3279    0.6074    0.4367    0.8414    0.7621
```

```
disp(['Rank of A: ', num2str(rankA)]);
```

```
Rank of A: 7
```

```
Reduced_rank_of_A = rref(A)
```

```
Reduced_rank_of_A = 7×9
     1     0     0     0     0     0     1     0     0
     0     1     0     0     0     0     0     1     0
     0     0     1     0     0     0     0     0     1
     0     0     0     1     0     0     0     0     0
     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     0     0
```

```


% (b) Find the SVD of the matrix

[U,S,V] = svd(A);
disp('Matrix U: ');
```

```
Matrix U:
```

```
disp(U);
```

```
   -0.4502   -0.1179    0.1701   -0.0272    0.5055   -0.6834   -0.1767
   -0.4508   -0.5347   -0.2587    0.4324    0.0325    0.2129    0.4590
   -0.3031   -0.4119   -0.2826   -0.4470   -0.4845   -0.0179   -0.4730
   -0.2720    0.5067   -0.5041   -0.3535   -0.1022   -0.2197    0.4812
   -0.2616    0.2069    0.4545    0.3527   -0.6813   -0.2919    0.0916
   -0.3759    0.4680   -0.2810    0.4211    0.1164    0.3026   -0.5275
   -0.4676    0.1092    0.5336   -0.4311    0.1431    0.5121    0.1272
```

```
disp('Matrix S: ');
```

```
Matrix S:
```

```
disp(S);
```

```
    3.8918         0         0         0         0         0         0         0         0
         0    1.5086         0         0         0         0         0         0         0
         0         0    1.0847         0         0         0         0         0         0
         0         0         0    0.6455         0         0         0         0         0
         0         0         0         0    0.4801         0         0         0         0
         0         0         0         0         0    0.2262         0         0         0
         0         0         0         0         0         0    0.0000         0         0
```

```
disp('Matrix V: ');
```

```
Matrix V:
```

```
disp(V);
```

```
   -0.2705    0.4835   -0.1381    0.0729    0.1411   -0.3857   -0.0822    0.1884   -0.6766
   -0.2307    0.1827    0.6018   -0.1498   -0.1403    0.0957    0.2581   -0.6254   -0.2055
   -0.4228   -0.2152    0.0055    0.1149    0.4637    0.2160   -0.6531   -0.2709    0.0040
   -0.3569    0.2290   -0.4818   -0.4815   -0.3268    0.4994    0.0000    0.0000   -0.0000
   -0.3379   -0.0902   -0.0634    0.7379   -0.5684    0.0781    0.0000   -0.0000   -0.0000
   -0.3848   -0.5590   -0.0370   -0.3766   -0.2469   -0.5791    0.0000   -0.0000    0.0000
   -0.2705    0.4835   -0.1381    0.0729    0.1411   -0.3857    0.0822   -0.1884    0.6766
   -0.2307    0.1827    0.6018   -0.1498   -0.1403    0.0957   -0.2581    0.6254    0.2055
   -0.4228   -0.2152    0.0055    0.1149    0.4637    0.2160    0.6531    0.2709   -0.0040
```

```
% (c)
A = rand(7,9);
A(:,7:9) = A(:,1:3);
[U,~,V] = svd(A);

% orthonormal basis for column space
c_s_b = U(:,1:rank(A));
disp('Orthonormal basis for column space: ');
```

```
Orthonormal basis for column space:
```

```
disp(c_s_b);
```

```
   -0.3434    0.1071    0.6632    0.1664    0.4020   -0.2548
   -0.2268   -0.6656    0.1209   -0.2370   -0.3088    0.3989
   -0.3740    0.4645   -0.1938    0.4997   -0.2371    0.5041
   -0.3439   -0.4700   -0.5229    0.4283    0.3606   -0.2715
   -0.4416    0.1383   -0.1064   -0.5297    0.5158    0.3899
   -0.4118    0.2615   -0.2984   -0.4227   -0.3723   -0.5326
   -0.4559   -0.1463    0.3663    0.1545   -0.3900   -0.1119
```

```
% orthonormal basis for left null space
l_n_s_b = V(:,rank(A) + 1:end);
disp('Orthonormal basis for left null space: ');
```

```
Orthonormal basis for left null space:
```

```
disp(l_n_s_b);
```

```
    0.2836   -0.2675    0.5899
    0.0200   -0.6401   -0.2998
    0.6474    0.1369   -0.2492
   -0.0000   -0.0000    0.0000
   -0.0000   -0.0000   -0.0000
    0.0000    0.0000   -0.0000
   -0.2836    0.2675   -0.5899
   -0.0200    0.6401    0.2998
   -0.6474   -0.1369    0.2492
```

```
% (d)
% orthonormal basis for right singular space
r_s_s_b = V(:,1:rank(A));
disp('Orthonormality of Right Singular Space and Left Null Space: ');
```

```
Orthonormality of Right Singular Space and Left Null Space:
```

```
disp(r_s_s_b'* l_n_s_b);
```

```
   1.0e-15 *

    0.0555         0    0.0555
   -0.0278         0   -0.0416
   -0.1110    0.0555   -0.0208
    0.2914    0.1943    0.0555
    0.1249    0.1110    0.0208
         0   -0.0278   -0.0139
```

```
%(e)
A = rand(7,9);
A(:,7:9) = A(:,1:3);
[U,S,V] = svd(A);
disp('Singular values of A: ');
```

```
Singular values of A: 
```

```
disp(diag(S));
```

```
    4.1907
    1.3505
    1.2196
    0.8201
    0.2913
    0.1877
    0.0000
```