**ASSIGNMENT - 3**

INTRODUCTION TO COMPUTING

SUBMITTED BY ADITYA SINGH 2K19/EP/005

**Question 1**

a) Is there a rot180 function? Is there a rot-90 function (to rotate clockwise)? Write command to rotate a matrix 180 degrees and to rotate it 90 degrees clockwise.

b) Write a code that can be used for changing the dimensions.

Input matrix: 1 2 3 Output matrix: 1 4 2 5 3 6 4 5 6

4 5 6

c) Solve the given linear equations for v,w,x,y,z. Equations :

3v - 3w + 6x - 2y + z = 14

3v - 6w + x - y + z = 25

2v - 4w + 4x - 4y+ 3z = 5

3v - 6w + 5x - y + 2z = 30

2v - 4w + 9x + y + z = 30

d) Using the g and h given, try out the following.

g = [1 2 3 4; 5 6 7 8; 9 10 11 12]

h = [3 3 4 4; 5 5 6 6; 7 7 8 8]

i) h >= g

ii) g == h

**Solution**

a) NO, There is no such rot180 function. But there is rot90(A,k) that rotates array A counterclockwise by k\*90 degrees, where k is an integer.

A = [1 2 3; 4 5 6];

a1 = rot90(A,2) % Rotate 180 degrees

a2 = rot90(A,-1) % Rotate 90 degrees clockwise

b) b = reshape(A,1,6)

c) B= [3 -3 6 -2 1; 3 -6 1 -1 1; 2 -4 4 -4 3; 3 -6 5 -1 2; 2 -4 9 1 1]

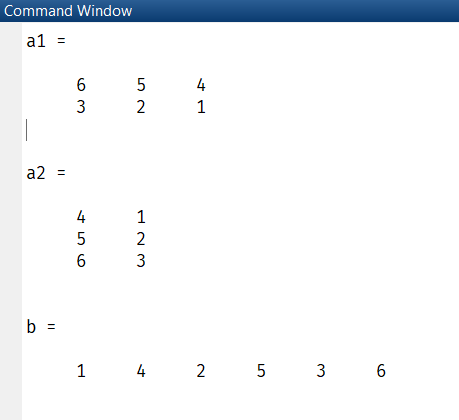
C = [14; 25; 5; 30; 30]

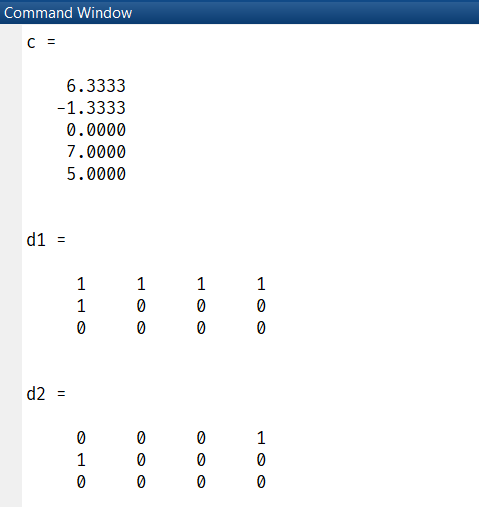
c = B\C

d) d1 = h >= g

d2 = g == h

**Output**





**Question 2**

a) Write matlab commands for the following :

• makes a n × n random matrix A normally distributed

• calculates its inverse B

• multiplies the two back together

• calculates the residual (i.e. difference between AB and identity matrix)

b) Create a row vector d with 5 linear spaced entries, say, containing the values of the diagonal entries (inorder) then create a required diagonal matrix. Also add vector [2, -5,7] to 2nd upper off diagonal elements.

**Solution**

1. a1 = rand(5)

a2 = inv(a1)

a3 = a1\*a2

a4 = a3-eye(5)

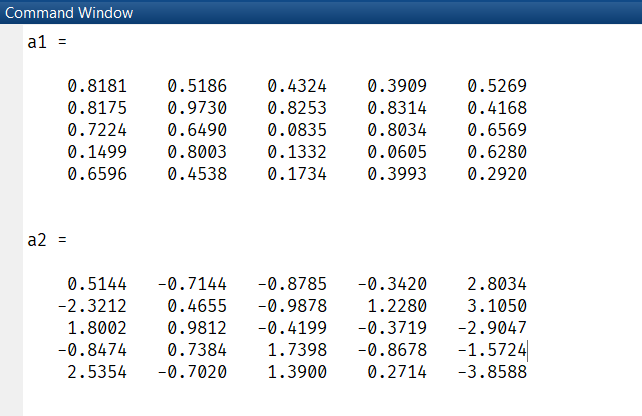
1. d = [5 6 2 4 3]

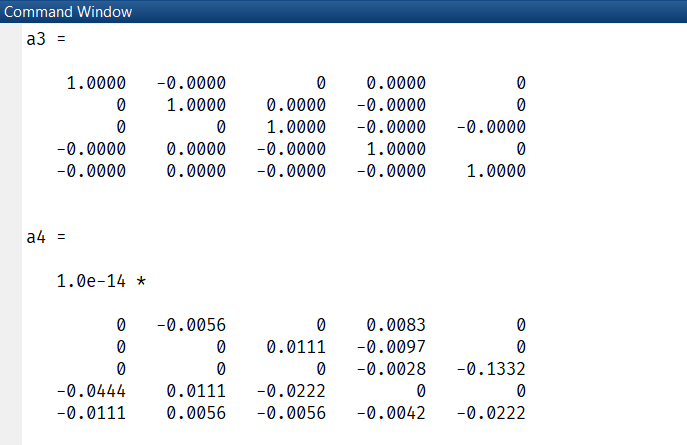
e = [2, -5,7]

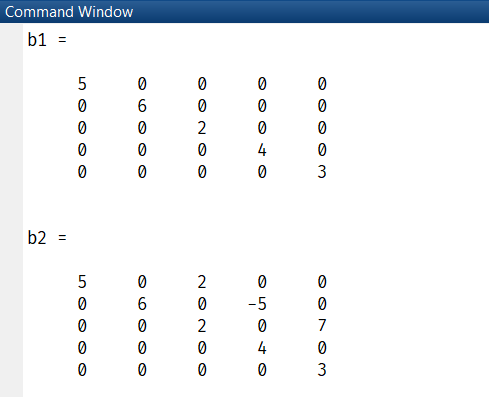
b1 = diag(d)

b2 = diag(d) + diag(e,2)

**Output**







**Question 3**

Let A = [1 7 3 5; 9 1 8 6; 2 0 3 0]. Explain the results and perform the following command:

i) compute the reciprocal of each element of A

ii) compute the square-root of each element of A

iii) swap the 2nd row and the last row

iv) obtain lower triangular elements of the matrix

v) [A A(end, :)]

vi) convert A into a 4-by-3 array

vii) add a row of all 1’s at the beginning and at the end

**Solution**

a = 1./A % reciprocal of each element

b = sqrt(A) % square root of each element

c = A; c([2,3],:) = c([3,2],:) % swaps 2nd and 4th row

d = tril(A) % lower triangular

e = [A A(end, :)] % error - Adding a row in 3x3 matrix

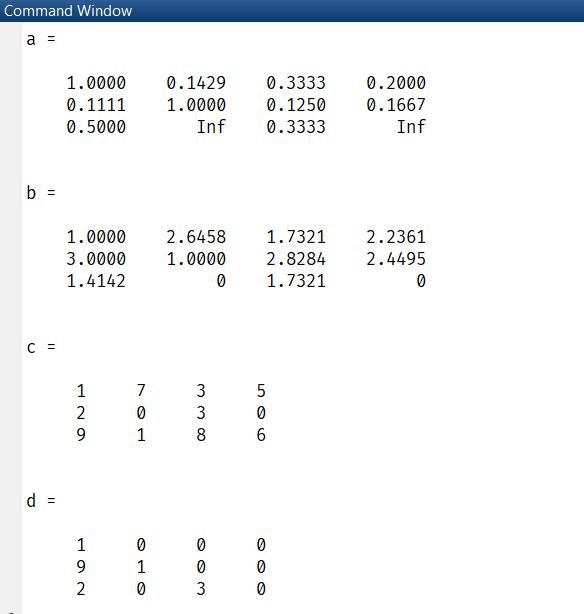
f = reshape(A,4,3) % reshape 4x3

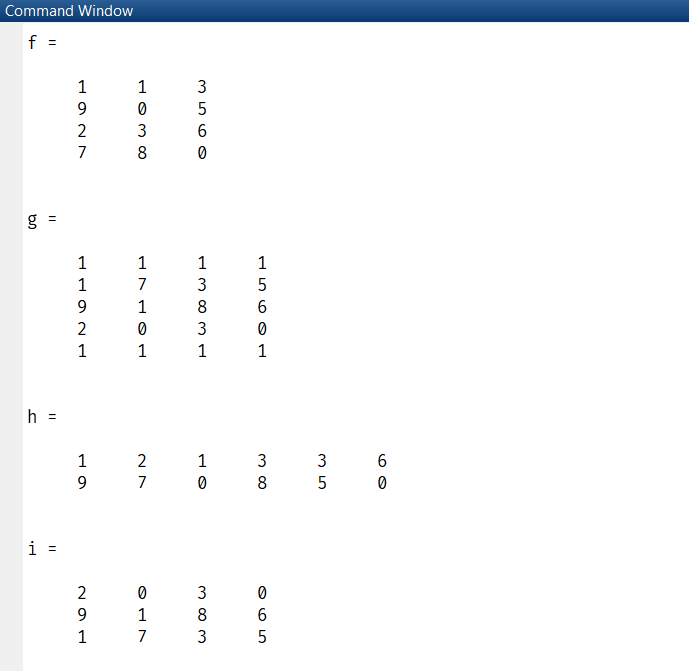
g = [1 1 1 1; A; 1 1 1 1] % add rows of 1 at beginning & end

h = reshape(A,2,6) % reshape 2x6

i = flipud(A) % flip along horizontal

**Output**





**END**