```
A = [2 3 4; 3 4 5]
B = [1 2 ; 12  3; 23 45]
C = A*B % matrix multipication of mxn and nxp matrix to generate mxp matrix
D = eye(2)

E = C.*D % element-wise multipication of mxn and mxn matrix to generate mxn matrix
E = E./16 % element-wise each element divided by 16
E = E.^2 % element-wise each element raised to power of 2
log(E) % element-wise
exp(E) % element-wise

a = 3.90
disp(a)
round(a)
disp(a)


m = max(A) % maximum element in each column
J = [2 3 4 5 6]
j  = max(J) % gives 6 because now there are no rows in each column to do column-wise max
J>3 % element-wise comparison 1 for true 0 for false
find(J>3) % shows all indexes where this is true

K = magic(3) % see help
L = eye(3)
T = [1 2 3; 3 4 5; 45 6 3]
r = max(T,K) % element-wise max between corresponding elements of two matrices

w1 = max(T,[],1) % same as max(T) % maximum element in each column
w2 = max(T,[],2) % maximum element in each row
w3 = max(max(T))
sum(T) % column-wise sum i.e. sum of all elements of a column so you will get 1(one)xn matrix
sum(T,1) %column-wise sum
sum(T,2) %row-wise sum
% now if you take sum of the ans above generated then you will get full sum of matrix
sum(sum(T)) % complete sum of all elements of T

Q = 10*rand(3)
floor_value = floor(Q) % gretest integer function applied element-wise
ceil_value = ceil(Q) % least integer function applied element-wise

inverse_of_matrix = inv(ceil_value)
pseudo_inv_of_matrix = pinv(ceil_value)
p2 = pinv(w1) % w1 is not a square matrix so its inverse doesn't exist but pseudo inverse does

```