```
% percep1e  executes 1 full epoch of PERCEPTRON LEARNING
% by calling function percept1p repeatedly
% It receives a matrix "P" with all the trainnig patterns
% as its columns AND a matrix "T" with the corresponding
% target vectors as its columns. It ALSO receives intial
% weights (matrix) and biases (vector)
% It returns the FINAL WEIGHT MATRIX and FINAL BIAS VECTOR
% after completion of the whole epoch of training, AS WELL AS
% the number of non-zero errors found in the epoch.
%
% SYNTAX: [Wepoch, bepoch, nze]=percep1e(W0,b0,P,T);
%
function [Wepoch, bepoch, nze]=percep1e(W0,b0,P,T);
% Find rows and cols of W (rows=#output PEs, cols=# output PEs)
[rw,cw] = size(W0);
% find # of cols in P to know how many patterns are in train set
[rp,cp] = size(P);
% Make W0 and b0 the "old" W and b, to get started
Wold = W0;
bold = b0;
% Initialize a counter for nonzero errors in the epoch to 0
nze = 0;
% FOR LOOP TO GRAB EACH PATTERN (col of P) at a time:
for pt = 1:cp
    % grab the one pattern form P and corresponding targ from T
    thisp = P(:,pt);
    thist = T(:,pt);
    % CALL percept1p for thisp and thist
    [Wnew, bnew,e] = percept1p(Wold, bold, thisp, thist);
    % INCREMENT nze ONLY IF CHANGES TOOK PLACE
    % if ((Wnew ~= Wold) | (bnew ~= bold) )
    if ( e ~= 0)
        nze = nze + 1;
    end
    % "pass" the new W and b as "old" W and b for next pattern
    Wold = Wnew;
    bold = bnew;
end   % END OF THE FOR LOOP that goes through all the patterns
Wepoch = Wnew;
bepoch = bnew;
end
```