Ex:

Write a Matlab® function called pos_match that accepts as an argument an array (of integers) called in_var and does the following:

i) Extracts values from the array that are equal to their linear index position in the array
ii) Returns the values extracted in step (i) in an array called out_var

For example, suppose we have

\[ A = [5, 1, 3; 4, 7, 8; 2, 6, 9]' \]

Then

\[ \text{pos_match}(A) \]

produces the following returned array:

\[
\begin{bmatrix}
3 \\
4 \\
9
\end{bmatrix}
\]

Note that we get this result because \( A(3) = 3 \), \( A(4) = 4 \), and \( A(9) = 9 \).

Sol'n:

```matlab
function out_var = pos_match(in_var)
% pos_match.m    Finds values in array that equal their linear index position.
%     % out_var = pos_match(in_var)
% i) ii) The compact (but hard to decipher) method:
% out_var = in_var(in_var(1:numel(in_var)) == 1:numel(in_var))';
% i) The easy to understand method.
% out_var = [ ];
for index = 1:numel(in_var) % or size(in_var,1) * size(in_var,2)
    if in_var(index) == index % linear indexing of in_var.
        out_var = [out_var; index]; % Note the ; so array vertical
    end
end
end
```