

**Topics:** Pre-allocating arrays, logical arrays, review

## Pre-allocating vectors vs. building vectors by concatenation

```
% Add vectors a and b of same length
n= length(a);
c= zeros(1,n); % unnecessary to pre-allocate, but ...
for k= 1:n
    c(k)= a(k) + b(k);
end
```

## Logical arrays and operations

Logical arrays, i.e., arrays containing logical values, are the results of *relational* or *logical* operations. In MATLAB, logical values are zero for false and one (or any non-zero value) for true. Logical values are not just numbers—they have the *logical property* attached to the data, see the workspace window under “class” when you have a logical value in the MATLAB workspace.

```
elev = 8*rand(4,3) + 10 % example, elevations on a map
elev > 16 % returns a logical array

% 1-d examples
vec = elev(1,:) % 1st row of matrix elev

L = vec>16 % logical array indicating result from vec>16
vecHigh = vec(L) % extract just the cells with values > 16

vecHigh = vec(vec>16) % combine last two statements in one
% this shortcut works for VECTORS only, not matrices

ind = find(vec>16) % get the indices where vec>16
vecHigh = vec(ind) % extract just the cells with values > 16

% Create a vector same as vec above except that all the values below 16 are "zeroed out".
L = (vec>16) % a LOGICAL vector
vecHigh = zeros(1,length(vec))
vecHigh(L) = vec(L) % assign only to the cells with logical value 1

ind = find(vec>16) % a vector of INDICES
vecHigh = zeros(1,length(vec))
vecHigh(ind) = vec(ind) % assign only to the cell numbers stored in ind

% There is a simpler solution using vectorized multiplication...

% 2-d examples
L = elev>16 % logical array (matrix)
elevHigh = elev(elev>16) % a VECTOR!!!

[ri,ci] = find(elev>16) % ri is vector that stores row index where elev>16
% ci is vector that stores col index where elev>16
```

## Another simulation example

Simulate the rolling of a *fair* die. The function below allows the user to specify the number of rolls. Be careful about using the random number generator for generating integers *with equal probability*.

```
function count = rollDie(rolls)
% Simulate rolling of fair 6-sided die
% Usage: count = rollDice(rolls)
% ROLLS is the number of times to roll die
% COUNT is vector of how many times each outcome occurs
% count(f) is the number of times face f occurs

FACES= 6; % number of faces on die

count= % bins to store counts

% Count outcomes of rolling a FAIR die
for k= 1:rolls

    face=

end

% Show histogram of outcome
bar(1:FACES, count);
title(['Outcomes from ' num2str(rolls) ' rolls of fair die']);
xlabel('Outcome'); ylabel('Count');
```

## Example: Reflect a matrix across its diagonal

```
% Given SQUARE matrix M, reflect the lower triangular portion to the upper triangular portion
% and calculate the sum of the lower (or upper) triangular portion, excluding the diagonal.
% Example:  2 1 6 9      2 5 4 7
%           5 1 4 0  --> 5 1 3 2
%           4 3 2 8      4 3 2 1
%           7 2 1 8      7 2 1 8
```