1  **Find a value in a matrix**

Write the following function:

```matlab
function [rvec, cvec] = findInMatrix(n,M)
% Find all occurrences of the number n in matrix M.
% rvec and cvec are column vectors of row and column numbers such that
% M(rvec(k),cvec(k)) is equal to n.
% If n is not found in M, rvec and cvec are empty vectors.

Use loops in this problem; *do not* use the built-in function `find`.
```

2  **Cumulative sums**

Write the following function:

```matlab
function A = matrixCSums(M)
% M is a numeric matrix and A has the same size as M.
% Each element in A is the sum of the corresponding element in M and all
% the elements above it. Example:
% M = [ 1 3; ... ]
% 4 5; ... then 5 8; ...
% -7 2] -2 10]
% Do NOT use any built-in functions other than size
```

**Note:** The next two questions require that you *design* solutions. Instead of giving you the specifications of a function, we are asking you to design a complete solution: you decide what functions and/or scripts are necessary and implement those functions/scripts. Take some time to do the planning—think about what values you need to keep track of and choose “appropriately-shaped” variables to store them.

3  **Random walk**

A random walk that starts from the center of a $21 \times 21$ grid ends when a boundary is reached. On average which “square” or grid point is visited most often? Function `RandomWalk2D` (discussed in lecture) is shown on the next page for your reference.

4  **Bounded random walk**

In a bounded random walk, a set number of steps are taken within a bounded area. For example, when the right boundary (excluding the corners) is reached, the next step can go left, up, or down only. Similarly, when a corner is reached, the next steps can be in two directions only. For a 100-step bounded random walk in a $21 \times 21$ grid, which “square” is visited most often?
function [x, y] = RandomWalk2D(N)
% Simulate a 2D random walk in an (2N+1)-by-(2N+1) grid.
% N is a positive integer.
% Walk starts from the middle and continues until the an edge, abs(N),
% is reached.
% x and y are row vectors with the property that (x(k),y(k)) is the
% location of the token after k hops, k=1:length(x).

% Initializations...
k=0; xc=0; yc=0;

% (xc,yc) is the location after k hops.
while abs(xc)<N && abs(yc)< N
  % Standing at (xc,yc), randomly select a step
  r= rand(1);
  if r < .25
    yc= yc + 1;  % north
  elseif r < .5
    xc= xc + 1;  % east
  elseif r < .75
    yc= yc -1;  % south
  else
    xc= xc -1;  % west
  end
  % Record location...
  k= k + 1;  x(k)= xc;  y(k)= yc;
end