Topics: iteration using for, random number, review

Reading (ML): Sec 4.2–4.5 (exclude 4.2.2)

A random example. I mean a random walk example!

Write a program that performs a “random walk.” In a random walk, possible moves are left, right, up, or down (in a Cartesian plane). Prompt the user for the number of steps and the starting point.

% Perform n steps of random walk starting from (x0,y0)

disp('Do a random walk!')
n= input('How many steps? ');
x0= input('From what x-coordinate? ');
y0= input('From what y-coordinate? ');

% possible movements: ( xdir(i), ydir(i) )
  xdir= [ 1 -1 0 0];
  ydir= [ 0 0 1 -1];

x= [x0 zeros(1,n)]; % trajectory in x direction
y= [y0 zeros(1,n)]; % trajectory in y direction

% Perform walk, each step is based on a random integer

% Show the walk, starting point, ending point
plot(x,y,x(1),y(1),'*',x(end),y(end),'o')
title([num2str(n) ' steps of random walk from * to o'])
Building vectors through concatenation

% Add vectors a and b
n= length(a);
c = zeros(1,n); % unnecessary statement but improves performance
k= 1; while ( k <= n)
    c(k)= a(k) + b(k);
    k= k + 1;
end

Example 2

Write a program segment that determines whether a given integer n is prime. Assume n>2. (Hint: MATLAB function mod(x,y) returns the value of the remainder of x divided by y assuming integer values of x, y.)

Example 3

Sketch a program that will list all the prime numbers in the range of [a,b] given integers a,b>1 and a < b.

Example 4

Develop an algorithm for calculating the mode of a sequence. The mode is the number in the sequence that occurs with maximum frequency. Assume that the sequence is (a) non-negative, (b) entered one by one and terminated by a negative number, and (c) entered in non-decreasing order. E.g., the mode of the sequence 87,92,92,98,98,98,100 is 98. Assume that only scalar variables are allowed.

Programming Rules of Thumb

- **Learn program patterns** of general utility and use relevant pattern for the problem at hand.
- **Seek inspiration** by systematically working test data by hand. Be introspective; ask yourself: “what am I doing?”
- **Declare variables** for each piece of information you maintain when working problem by hand. Write comments that precisely describe the contents of each variable.
- **Decompose** problem into manageable tasks.
- **Refine** the algorithm iteratively: solve a simpler problem first
- **Remember** the problem’s boundary conditions.
- **Validate** your program by tracing it on simple test data.