

```

function y = recur(a,b,n,x,x0,y0);
%
% y = recur(a,b,n,x,x0,y0)
% solves for y[n] from:
%  $y[n] + a_1*y[n-1] + a_2*y[n-2] \dots + a_n*y[n-N]$ 
%  $= b_0*x[n] + b_1*x[n-1] + \dots + b_m*x[n-M]$ 
%
% a, b, n, x, x0 and y0 are vectors
% a = [a1 a2 ... aN]
% b = [b0 b1 ... bM]
% n contains time values for which solution will be computed
% x contains the input values over the range of indices n
% y0 contains the initial conditions for y, in order,
%   i.e., y0 = [y[n0-N], y[n0-N+1], ..., y[n0-1]]
%   where n0 represents the first element of n
% x0 contains the initial conditions on x, in order
%   i.e., x0 = [x[n0-M], ..., x[n0-1]]
% the output, y, has length(n)
%
N = length(a);
M = length(b)-1;
if length(y0) ~= N,
    error('Lengths of a and y0 must match')
end
if length(x0) ~= M,
    error('Length of x0 must match length of b-1')
end
y = [y0 zeros(1,length(n))];
x = [x0 x];
a1 = a(length(a):-1:1);    % reverses the elements in a
b1 = b(length(b):-1:1);
for i=N+1:N+length(n),
    y(i) = -a1*y(i-N:i-1)' + b1*x(i-N:i-N+M)';
end
y = y(N+1:N+length(n));

```