```
% Exercise 4.6 - the gamblers problem.
%
% See the Sutton book.
%
% Written by:
% --
% John L. Weatherwax                2007-12-03
%
% email: wax@alum.mit.edu
%
% Please send comments and especially bug reports to the
% above email address.
%
%-----

clc;
close all;

% we have an undiscounted task:
gamma = 1;

% the number of states (0 and 100 are terminal states)
n_non_term_states=99;
%n_non_term_states=1022;
n_states=n_non_term_states+2;

% initialize state value function (including terminal states):
V = zeros(1,n_states); V(1)=0.0; V(end)=1.0;

% the probability our coin lands heads:
%p_heads = 0.3;
p_heads = 0.40;

thetaThreshold = 1e-8;

% plot these iterations:
plotIters = [ 1, 2, 3, 32 ];

delta = +Inf; iterCnts = 0;
fhs=figure; hold on; grid on;
%fhp=figure; hold on; grid on;
while( delta > thetaThreshold )
  iterCnts=iterCnts+1;

  delta = 0;
  % loop over all NON TERMINAL states:
  for si=2:n_states-1,
    v = V(si);
    s = si-1;  % the state \in [1,\cdots,99]
    % get the possible actions in this state (not lower bound of ONE ... zero seems like
a unreasonable action)
    acts = 1:min(s,(n_states-1)-s); Q = [];

    for ai=1:length(acts),
      Q(ai) = gam_rhs_state_bellman(s,acts(ai),V,gamma,p_heads);
    end % end action loop
    V(si) = max(Q);
    delta = max(delta,abs(v-V(si)));
  end % end state loop
```

```
  if( isempty( setdiff( [iterCnts], plotIters ) ) )
    figure(fhs); plot( 0:(n_states-1), V, '-x' ); axis tight; drawnow;
  end
end
figure(fhs); plot( 0:(n_states-1), V, '-go' ); xlabel('capital'); ylabel('value ests.');
axis tight; drawnow;
fn='gam_state_value_fns.eps'; saveas( gcf, fn, 'epsc' );

% compute the greedy policy at each timestep:
%
eps_pol = 1e-8;
%eps_pol = 0;
pol_pi  = zeros(1,n_states-2);
% loop over all non-terminal states:
for si=2:n_states-1,
  s = si-1;  % the state \in [1,\cdots,99]
  % get the possible actions in this state (no zero action)
  acts = 1:min(s,(n_states-1)-s);
  %acts = fliplr(acts); % try to reverse the order of the actions presented and see if we
get a different policy

  Q = []; bestVal=-Inf; bestAct=0;
  for ai=1:length(acts),
    Q(ai) = gam_rhs_state_bellman(s,acts(ai),V,gamma,p_heads);
    %--
    % assume that we have to beat an earlier policy by at least eps_pol
    % this seems to encourage plays with the smallest bets
    %--
    if( bestVal<(Q(ai)-eps_pol) )
      bestVal=Q(ai);
      bestAct=ai;
    end
  end % end action loop

  pol_pi(si-1) = bestAct;

end % end state loop
if( 1 ) %isempty( setdiff( [iterCnts], plotIters ) ) )
  figure; stairs( 1:(n_states-2), pol_pi ); xlabel('capital'); ylabel('last policy');
axis tight; drawnow;
  fn='gam_final_policy.eps'; saveas( gcf, fn, 'epsc' );
end

return;
```