```
function v = stationary(P)
%
% This is a MATLAB function that calculates the stationary probability vect↙
or v
% of a Markov chain transition matrix P, i.e., we solve v = v*P .
% We assume the existence of a unique stationary vector.
% For a finite-state Markov chain, the condition is that the chain be irred↙
ucible.
%
% To solve the system of n equations in n unknowns, we use the matrix inver↙
se function inv.
% For a square matrix A, inv(A)*A = I, where I is the identity matrix (1's ↙
on the diagonal, 0's elsewhere).
%
% We input the matrix P when we call the function.
% First find the number n of rows in the transition matrix P.
s = size(P);
n = s(1);
% There is one redundant equation in the n equations v = vP.
% We fill gap by using the fact that v(1) + ... + v(n) = 1.
% We eliminate redundant equation by replacing last column of P with ones.
PP = P;
PP(:,end) = [];
w = ones(n,1);
PP = [PP w];
% PP is the matrix P with the last column replaced by a column of 1's.
%
% Note that for the desired v, v*PP equals v except the last element is 1.
% We thus need to modify the equation.
% For that purpose, introduce auxiliary matrices I and L.
% I is the identity matrix.
% L is all zeros except a 1 in the bottom right.
I = eye(n);
f = [zeros(1, n-1) 1];
L=diag(f);
% Now v = vP for prob vector v becomes:  v*(PP) = v + f*(1-v(n))
% Or, equivalently, v*(PP) = v*I - v*L + f
% Or   v*(PP - I + L) = f
%
% We want to solve v*R = f, where R = PP-I+L.
%
R = PP-I+L;
%
% We can solve v*R = f in three ways:
% First, we can write v = f/R, understanding v and f to be row vectors.
% Second, we can take transposes and work with column vectors.
% We get (v*R)' = R'*v' = f'.
% We then write v' = R'\f'.
% Third, we can solve for v by directly inverting the matrix R = PP-I+L.
```

```
% The solution is  v = f*RR, where RR is the inverse of R.
%
RR= inv(R);
v = f*RR;
%
% The desired v is the last row of the matrix RR.
% By multiplying RR by f, we get the last row of RR.
```