```
function y=country1(input2)

% country-1 procedure
% computes the two equations of country 1 given
% d and teta1star - inputs to be solved for
% beta, integralpoints, integralstep - globals produced by main program
% n11, n21, r1 - globals for country-2 procedure
% flagz - global: stores error warnings from solving country-2 for the main program
% options, start2 - global from main program with instructions for the solver


global beta n11 n21 r1 flagz
global options start2
global integralpoints integralstep

d = input2(1);
teta1star = input2(2);

%  external integral over n11 for country-1 agents

welfarez = zeros(1,integralpoints);        % this vector will store the integrands

for n11index = 1:integralpoints;

   n11 = (n11index-1)*integralstep;
   n21 = min( 1 , max(0,n11-d) );
   r1 = max( 0 ,  R( teta1star , ((1+beta)/2)*n11+((1-beta)/2)*n21 ) );

   % solving for teta2star
   [solution, funcvalue, reportcode,output] = fsolve(@country2, start2, options);
   teta2star = solution(4);
   flagz(1,n11index) = reportcode;
   teta2starindex = floor(teta2star/integralstep);
   integercorrection = teta2star/integralstep-teta2starindex;

   % internal integral over teta2 for country-1 agents

    % compute for all teta2 as if there are no runs (saves run time by using vector operations)
    teta2z = 0:integralstep:1;
    r2z = max(0,R(teta2z,0));
    ustayz = u( (1+beta)*r1+(1-beta)*r2z );
    urunz = u( (1+beta)*1+(1-beta)*r2z );
    % remember values at discontinuity point for correction later
    ustaymid = ustayz(teta2starindex);
    urunmid = urunz(teta2starindex);
    % run 2 area (overwrite)
    ustayz(1:teta2starindex) = u( (1+beta)*r1+(1-beta)*1 );
    urunz(1:teta2starindex) = u( (1+beta)*1+(1-beta)*1 );
    % correct for the approximation in the integral at the discontinuity
    ustayz(teta2starindex) = integercorrection*ustayz(teta2starindex)+(1-integercorrection)*ustaymid;
    urunz(teta2starindex) =  integercorrection*urunz(teta2starindex)+(1-integercorrection)*urunmid;
    % integrate
    welfarez(n11index) = mean(ustayz-urunz);

end

country1eqn = mean(welfarez);      % equation for country-1 agents


%  external integral over n21 for country-2 agents

for n21index = 1:integralpoints

   n21 = (n21index-1)*integralstep;
   n11 = min( 1 , max(0,n21+d) );
   r1 = max( 0 ,  R( teta1star , n11*((1+beta)/2)+n21*((1-beta)/2) ) );

   % solving for teta2star
   [solution, funcvalue, reportcode] = fsolve(@country2, start2, options);
   teta2star = solution(4);
```

```
    flagz(2,n21index) = reportcode;
    teta2starindex = floor(teta2star/integralstep);
    integercorrection = teta2star/integralstep-teta2starindex;

    % internal integral over teta2 for country-1 agents

     % compute for all teta2 as if there are no runs (saves run time by using vector operations)
     teta2z = 0:integralstep:1;
     r2z = max(0,R(teta2z,0));
     ustayz = u( (1-beta)*r1+(1+beta)*r2z );
     urunz = u( (1-beta)*1+(1+beta)*r2z );
     % remember values at discontinuity point for correction later
     ustaymid = ustayz(teta2starindex);
     urunmid = urunz(teta2starindex);
     % run 2 area (overwrite)
     ustayz(1:teta2starindex) = u( (1-beta)*r1+(1+beta)*1 );
     urunz(1:teta2starindex) = u( (1-beta)*1+(1+beta)*1 );
     % correct for the approximation in the integral at the discontinuity
     ustayz(teta2starindex) = integercorrection*ustayz(teta2starindex)+(1-integercorrection)*ustaymid;
     urunz(teta2starindex) = integercorrection*urunz(teta2starindex)+(1-integercorrection)*urunmid;
     % integrate
     welfarez(n21index) = mean(ustayz-urunz);

end

country2eqn = mean(welfarez);      % equation for country-2 agents

y = [ country1eqn  country2eqn ];
```