ii) This is the second file – anisodiff.m

```
function diff_im = anisodiff(im, num_iter, delta_t, kappa, option)
fprintf('Removing noise\n');


fprintf('Filtering Completed !!');

% Convert input image to double.
im = double(im);

% PDE (partial differential equation) initial condition.
diff_im = im;

% Center pixel distances.
dx = 1;
dy = 1;
dd = sqrt(2);

% 2D convolution masks - finite differences.
hN = [0 1 0; 0 -1 0; 0 0 0];
hS = [0 0 0; 0 -1 0; 0 1 0];
hE = [0 0 0; 0 -1 1; 0 0 0];
hW = [0 0 0; 1 -1 0; 0 0 0];
hNE = [0 0 1; 0 -1 0; 0 0 0];
hSE = [0 0 0; 0 -1 0; 0 0 1];
hSW = [0 0 0; 0 -1 0; 1 0 0];
hNW = [1 0 0; 0 -1 0; 0 0 0];

% Anisotropic diffusion.
for t = 1:num_iter

    % Finite differences. [imfilter(.,.,'conv') can be replaced by conv2(.,.,'same')]
    nablaN = imfilter(diff_im,hN,'conv');
    nablaS = imfilter(diff_im,hS,'conv');
    nablaW = imfilter(diff_im,hW,'conv');
    nablaE = imfilter(diff_im,hE,'conv');
    nablaNE = imfilter(diff_im,hNE,'conv');
    nablaSE = imfilter(diff_im,hSE,'conv');
    nablaSW = imfilter(diff_im,hSW,'conv');
    nablaNW = imfilter(diff_im,hNW,'conv');

    % Diffusion function.
    if option == 1
       cN = exp(-(nablaN/kappa).^2);
       cS = exp(-(nablaS/kappa).^2);
       cW = exp(-(nablaW/kappa).^2);
       cE = exp(-(nablaE/kappa).^2);
```

```
    cNE = exp(-(nablaNE/kappa).^2);
    cSE = exp(-(nablaSE/kappa).^2);
    cSW = exp(-(nablaSW/kappa).^2);
    cNW = exp(-(nablaNW/kappa).^2);
  elseif option == 2
    cN = 1./(1 + (nablaN/kappa).^2);
    cS = 1./(1 + (nablaS/kappa).^2);
    cW = 1./(1 + (nablaW/kappa).^2);
    cE = 1./(1 + (nablaE/kappa).^2);
    cNE = 1./(1 + (nablaNE/kappa).^2);
    cSE = 1./(1 + (nablaSE/kappa).^2);
    cSW = 1./(1 + (nablaSW/kappa).^2);
    cNW = 1./(1 + (nablaNW/kappa).^2);
  end

  % Discrete PDE solution.
  diff_im = diff_im + ...
        delta_t*(...
        (1/(dy^2))*cN.*nablaN + (1/(dy^2))*cS.*nablaS + ...
        (1/(dx^2))*cW.*nablaW + (1/(dx^2))*cE.*nablaE + ...
        (1/(dd^2))*cNE.*nablaNE + (1/(dd^2))*cSE.*nablaSE + ...
        (1/(dd^2))*cSW.*nablaSW + (1/(dd^2))*cNW.*nablaNW );



end
```