```matlab
%Floyd-Steinberg
function y = fs(x,k)
height = size(x,1);
width = size(x,2);
ed = [0 0 0 7 0;0 3 5 1 0;0 0 0 0 0]/16;
y = uint8(zeros(height,width));
z = zeros(height+4,width+4);
z(3:height+2,3:width+2) = x;
for i = 3:height+2,
for j = 3:width+2,
quant = floor(255/(k-1))*floor(z(i,j)*k/256);
y(i-2,j-2) = quant;
e = z(i,j)-quant;
z(i:i+2,j-2:j+2) = z(i:i+2,j-2:j+2)+e*ed;
end
end
end
```

```matlab
input_image_rgb = imread('Digital Image Processing/Assignment 2/vase.jpg');
input_image_gray=rgb2gray(input_image_rgb);
imshow(input_image_gray);
title('Input Gray Image');
```


Input Gray Image

```matlab
floyd= fs(input_image_gray,2);
imshow (floyd);
title('Floyd-Steinberg Dithering With 2 Levels')
```
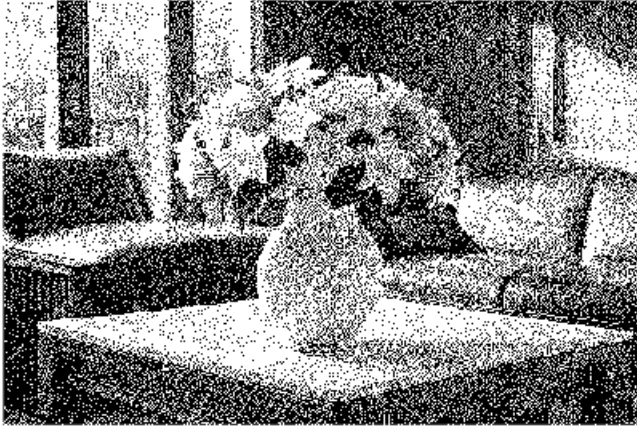
Floyd-Steinberg Dithering With 2 Levels


```
%Jarvis-Judice-Ninke with to levels
function out = jjn(im)
height = size(im,1);
width = size(im,2);
out = zeros(size(im));
ed = [0 0 0 7 5;3 5 7 5 3;1 3 5 3 1]/48;
z = zeros(size(im)+4);
z(3:height+2,3:width+2) = double(im);
for i = 3:height+2,
for j = 3:width+2,
quant = 255*(z(i,j)>=128);
out(i-2,j-2) = quant;
e = z(i,j)-quant;
z(i:i+2,j-2:j+2) = z(i:i+2,j-2:j+2)+e*ed;
end
end
out = im2uint8(out);
end
```

```
jarvis = jjn(input_image_gray);
imshow(jarvis);
title('Jarvis-Judice-Ninke Dithering With Two Levels');
```

Jarvis-Judice-Ninke Dithering With Two Levels

Comparison:

I chose the 2 levels of dithering for comparison of both results.

1. Floyd-Steinberg Method:

- The results show a sharper, grainy texture in the dithered image.
- Intensity changes are less distinguishable, especially in areas with smooth transitions.
- This method tends to introduce visible artifacts, which can make the image look harsher.

1. Jarvis-Judice-Ninke Method:

- The results produce a smoother image with less graininess.
- Intensity changes are easier to distinguish, providing a more natural and gradual transition between different shades.
- The method reduces artifacts, resulting in an overall higher quality image