Question 2

```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt

# General function to apply error diffusion based on a matrix
def error_diffusion_dither(image, diffusion_matrix, divisor):
    img = np.array(image, dtype=float) / 255.0  # Normalize image to
0-1 range
    rows, cols = img.shape
    height, width = len(diffusion_matrix), len(diffusion_matrix[0]) //
2  # Define height and half-width of the matrix

    for y in range(rows):
        for x in range(cols):
            old_pixel = img[y, x]
            new_pixel = round(old_pixel)
            img[y, x] = new_pixel
            quant_error = old_pixel - new_pixel

            # Apply diffusion to surrounding pixels
            for dy in range(height):
                for dx in range(-width, width + 1):
                    ny, nx = y + dy, x + dx
                    if 0 <= ny < rows and 0 <= nx < cols:
                        img[ny, nx] += quant_error *
diffusion_matrix[dy][dx + width] / divisor

    return (img * 255).astype(np.uint8)

# Floyd-Steinberg specific diffusion matrix
def floyd_steinberg_dither(image):
    floyd_steinberg_matrix = [
        [0, 0, 7],
        [3, 5, 1]
    ]
    return error_diffusion_dither(image, floyd_steinberg_matrix,
divisor=16)

# Jarvis-Judice-Ninke specific diffusion matrix
def jarvis_judice_ninke_dither(image):
    jjn_matrix = [
        [0, 0, 0, 7, 5],
        [3, 5, 7, 5, 3],
        [1, 3, 5, 3, 1]
    ]
    return error_diffusion_dither(image, jjn_matrix, divisor=48)
```

```
# Load an image (grayscale)
image_path = 'image_bmp.bmp'  # Replace with your image path
image = Image.open(image_path).convert('L')  # Convert to grayscale

# Apply Floyd-Steinberg and Jarvis-Judice-Ninke dithering
floyd_steinberg_result = floyd_steinberg_dither(image)
jarvis_judice_ninke_result = jarvis_judice_ninke_dither(image)

# Plot the original and dithered images
plt.figure(figsize=(12, 6))
plt.subplot(1, 3, 1)
plt.imshow(image, cmap='gray')
plt.title("Original Image")
plt.axis('off')

plt.subplot(1, 3, 2)
plt.imshow(floyd_steinberg_result, cmap='gray')
plt.title("Floyd-Steinberg Dithered")
plt.axis('off')

plt.subplot(1, 3, 3)
plt.imshow(jarvis_judice_ninke_result, cmap='gray')
plt.title("Jarvis-Judice-Ninke Dithered")
plt.axis('off')

plt.tight_layout()
plt.show()
```