```
import cv2
import numpy as np
import matplotlib.pyplot as plt
# Load image in grayscale
image = cv2.imread('lenagray.jpg', cv2.IMREAD_GRAYSCALE)
if image is None:
    print("Image not found")
    exit()
# 1. GLCM

def compute_glcm(img, distance=1, angle=0):
    rows, cols = img.shape
    levels = 256
    glcm = np.zeros((levels, levels), dtype=np.float64)

    dx = distance
    dy = 0

    for i in range(rows - dy):
        for j in range(cols - dx):
            row_val = img[i, j]
            col_val = img[i + dy, j + dx]
            glcm[row_val, col_val] += 1

    glcm /= glcm.sum()
    return glcm

glcm = compute_glcm(image)
```

```
# Extract GLCM Features
contrast = 0
energy = 0
homogeneity = 0
correlation = 0

mean_i = np.sum(np.arange(256) * np.sum(glcm, axis=1))
mean_j = np.sum(np.arange(256) * np.sum(glcm, axis=0))

std_i = np.sqrt(np.sum(((np.arange(256) - mean_i) ** 2) * np.sum(glcm, axis=1)))
std_j = np.sqrt(np.sum(((np.arange(256) - mean_j) ** 2) * np.sum(glcm, axis=0)))

for i in range(256):
    for j in range(256):
        contrast += (i - j) ** 2 * glcm[i, j]
        energy += glcm[i, j] ** 2
        homogeneity += glcm[i, j] / (1 + abs(i - j))
        if std_i != 0 and std_j != 0:
            correlation += ((i - mean_i) * (j - mean_j) * glcm[i, j]) / (std_i * std_j)

print("GLCM Features:")
print("Contrast:", contrast)
print("Energy:", energy)
print("Homogeneity:", homogeneity)
print("Correlation:", correlation)

# 2. SIFT Feature Extraction
```

```
sift = cv2.SIFT_create()
keypoints, descriptors = sift.detectAndCompute(image, None)

print("\nSIFT Features:")
print("Number of Keypoints:", len(keypoints))
print("Descriptor Shape:", descriptors.shape)

sift_image = cv2.drawKeypoints(image, keypoints, None)

# Display Results
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(image, cmap='gray')
plt.title("Original Image")
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(sift_image, cmap='gray')
plt.title("SIFT Keypoints")
plt.axis('off')

plt.show()
```