**Quiz 1**

**Digital image processing Lab**

**CEL 444**

|  |  |
| --- | --- |
| Name:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Enrolment #:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Instructions:**

* You have **30 minutes** to complete and submit this quiz. Ensure you submit your work on the **LMS (Learning Management System)** before the deadline, as late submissions will not be accepted.
* Your submission must include your code and screenshots.
* The use of **AI tools or automated coding** is **strictly prohibited**. Any detection of AI-generated content will­­­ result in an **"F" grade for the entire course**.
* Ensure your code is properly commented and reflects your understanding of the concepts.

**Question:**

Write a Python program to manually binarize an image without using any built-in thresholding functions from OpenCV. Your program should load an image, apply a threshold value of 127, and display both the original and the binarized images.

*import* cv2

*import* numpy *as* np

*import* matplotlib.pyplot *as* plt

image = cv2.imread('random image.jpg', cv2.IMREAD\_GRAYSCALE)

height, width = image.shape

binarized\_image = np.zeros((height, width), dtype=np.uint8)

threshold = 127

*for* i *in* range(height):

*for* j *in* range(width):

*if* image[i, j] > threshold:

            binarized\_image[i, j] = 255

*else*:

            binarized\_image[i, j] = 0

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.title("Original Image")

plt.imshow(image, cmap='gray')

plt.subplot(1, 2, 2)

plt.title("Binarized Image")

plt.imshow(binarized\_image, cmap='gray')

plt.show()

