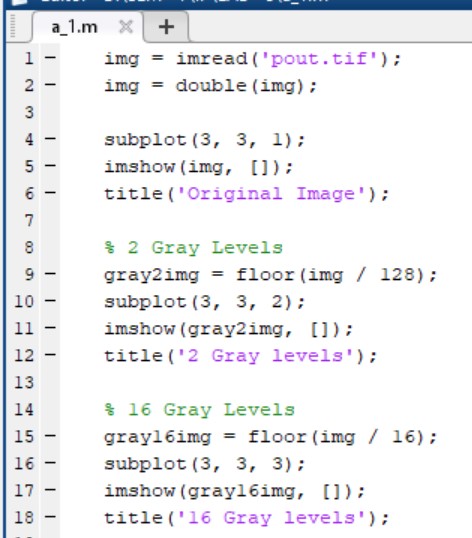
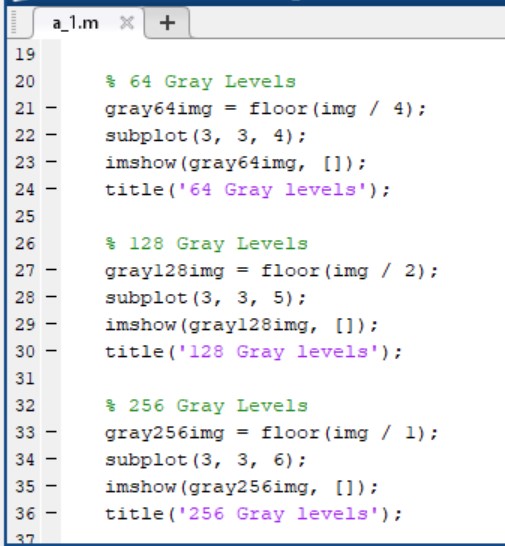
LAB - 5

|  |  |
| --- | --- |
| Name | Keval D Gandevia |
| Roll Number | CE046 |
| ID | 19CEUEG017 |
| Subject | Image Processing |

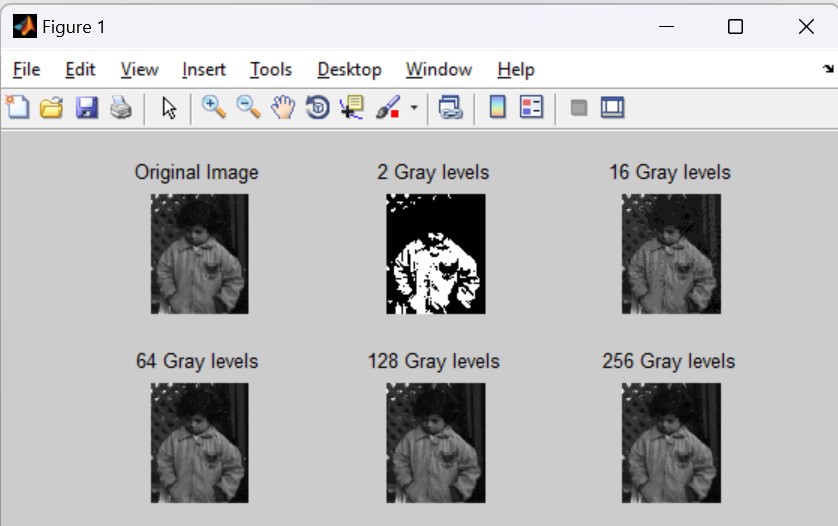
**Aim:** Implement the following algorithms.

**Q. 1: Take 'pout.tif' image and perform Gray-level Slicing on it. Diplay images with 2, 16, 64, 128 and 256 graylevels.**

* **Code:**

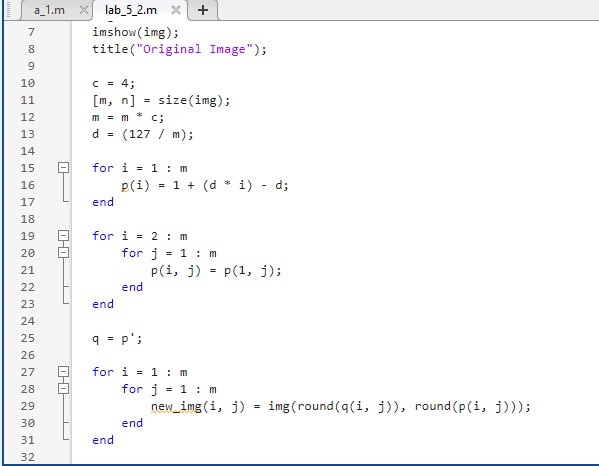


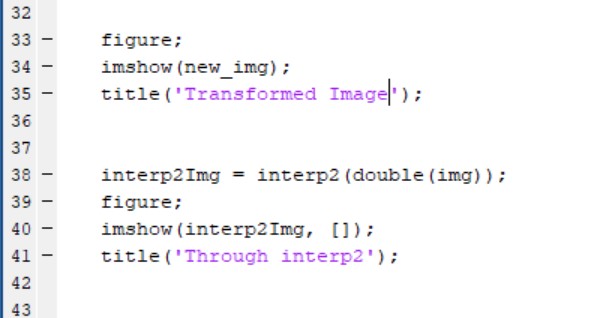
* **Output:**

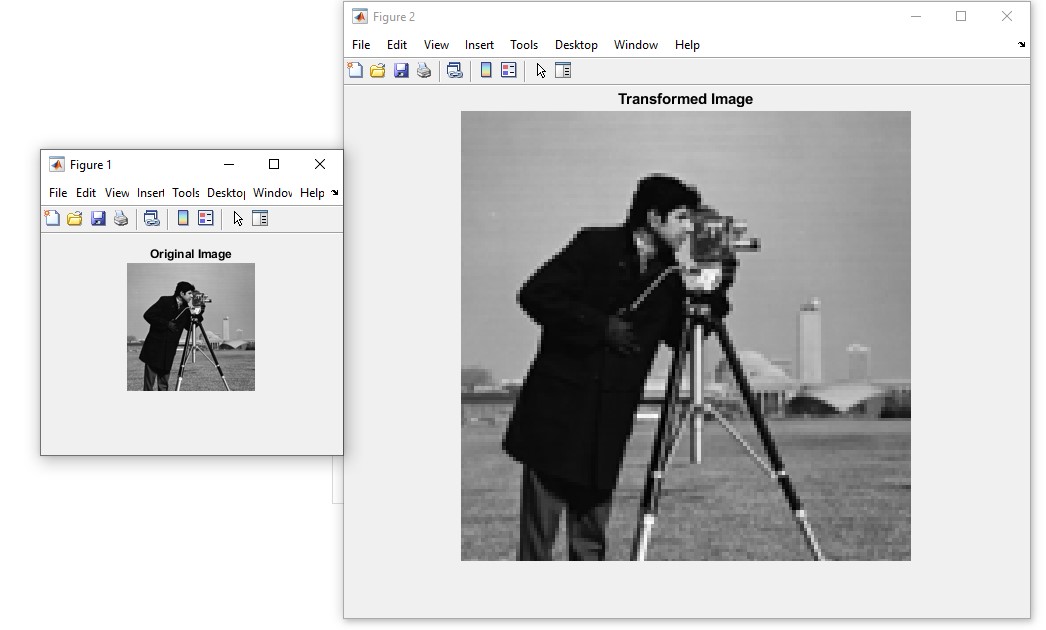
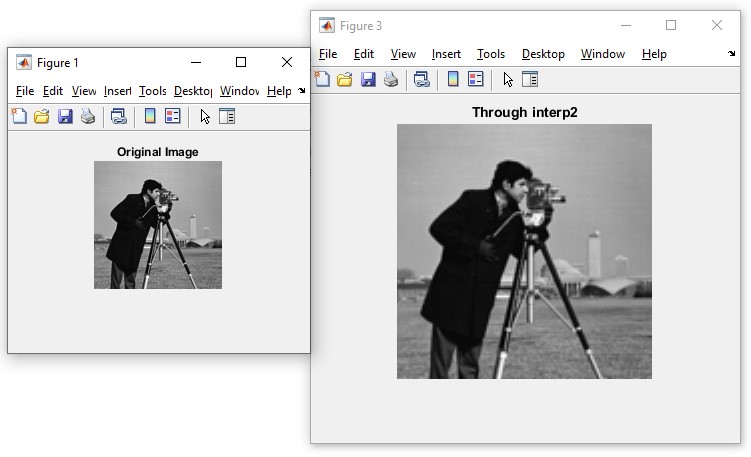


**Q. 2: Consider an image of 128x128 (Hint: You can resize 'cameraman.tif' to 0.5) and Implement Nearest-Neighbour Interpolation Algorithm and covert into 256x256. Don't use in-buit functions like linspace, meshgrid and interp2. Compare your result with the result obtained using the function interp2.**

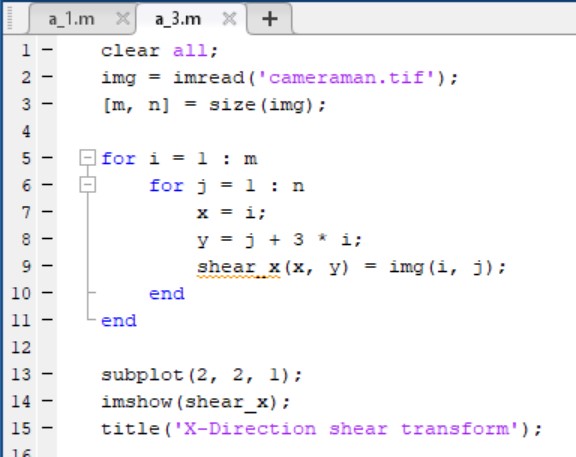
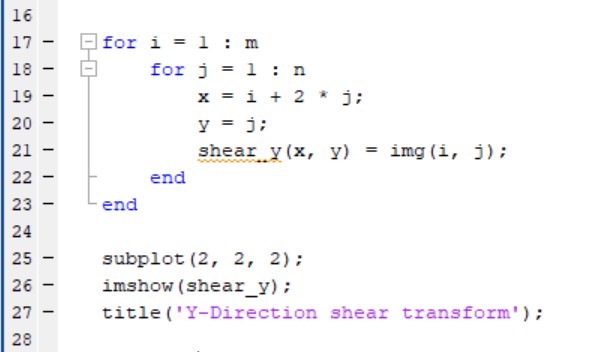
* **Code:**

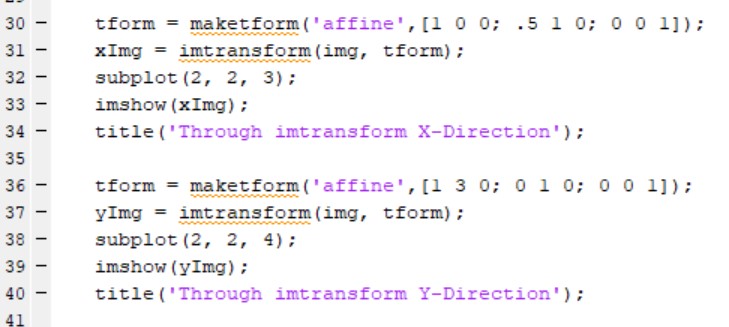




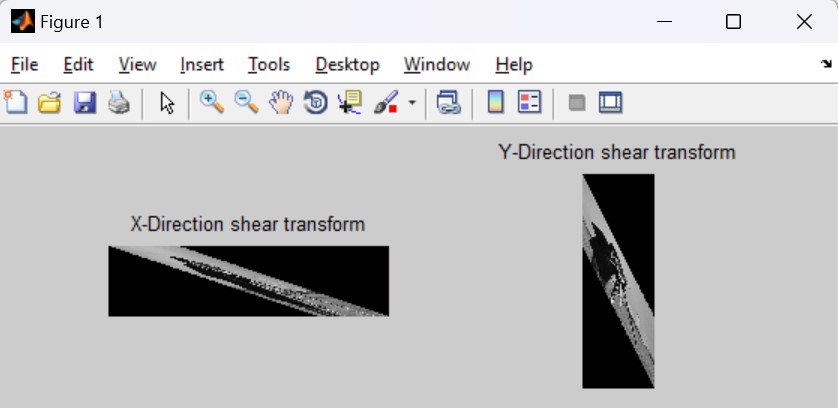
* **Output:**

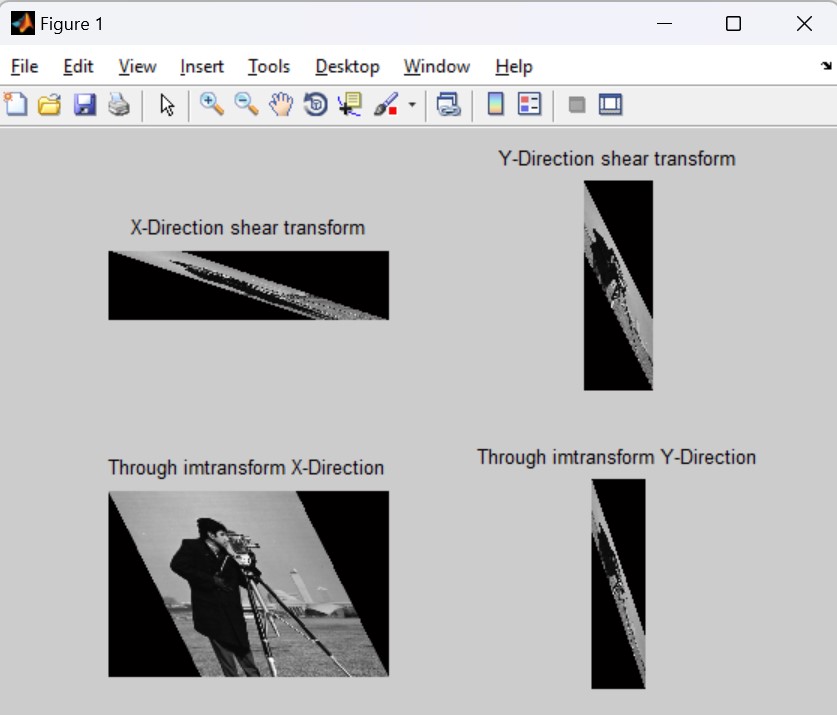
**Q. 3: Take 'Cameraman.tif' image and implement Shear Transformation. A). Apply shear transformation in X-direction with value 2. B). Apply shear transformation in Y-Direction with value 3. C). Compare your result with the output generated by in-built function imtransform.**

* **Code:**



* **Output:**





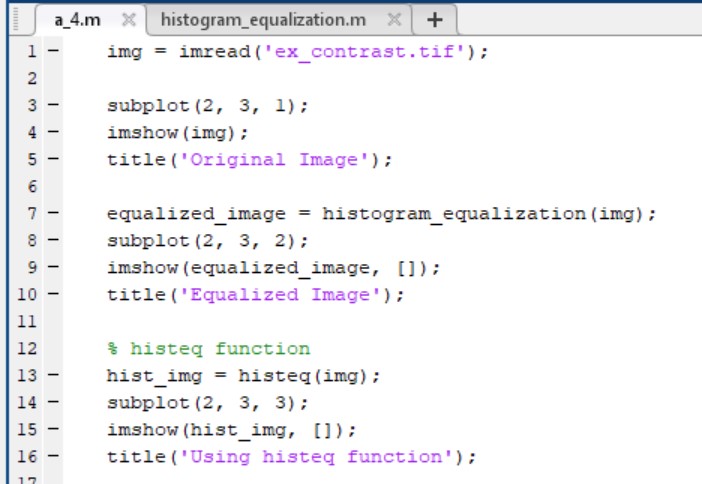
**Lab – 4**

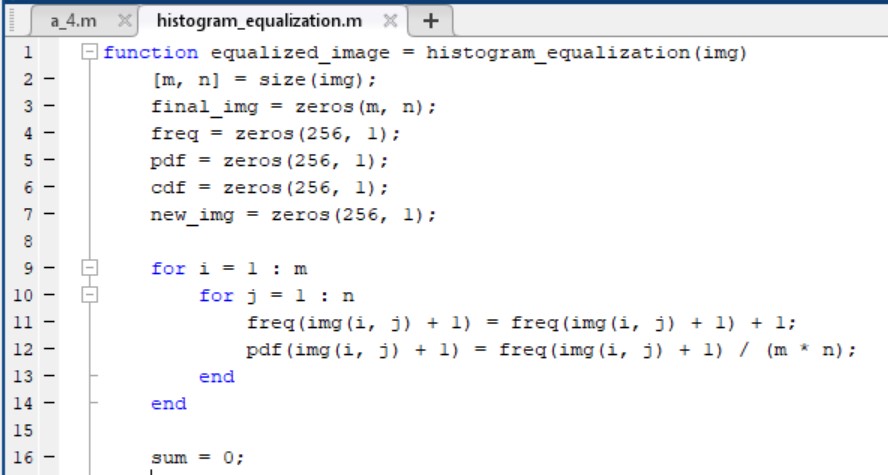
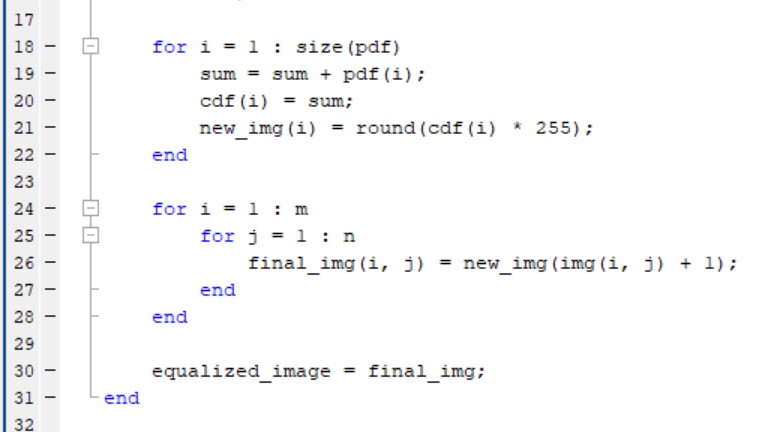
**Q. 4: Histogram Equalization. A). Create a function that would be able to perform histogram equalization on**

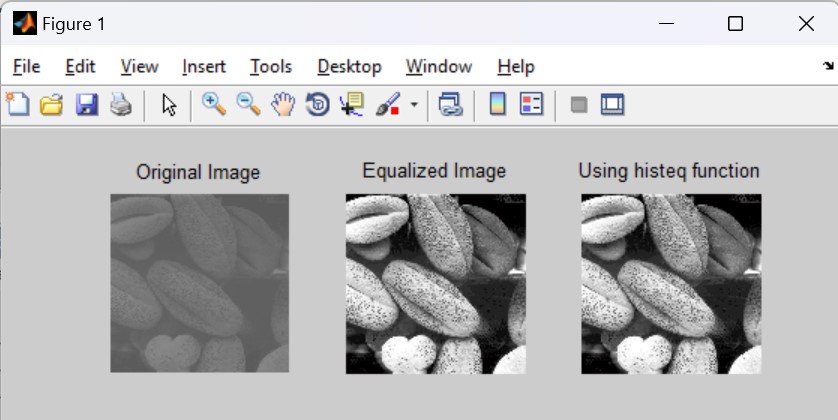
**a grayscale image. B). Use this function to equalize a low contrast image ex\_contrast.tif (from**

**Lab 2). C). Use the function histeq(image) on the same image ex\_contrat.tif. D). Compare the results of b) and c).**

* **Code:**





* **Output:**