Ex1:

1.

* First, the script reads a color image named sea.jpg.
* It extracts the three color channels: Red, Green, and Blue.
* Then, for each channel, it creates a new image where only that channel is kept, and the others are set to black (zero).
* The original image and the three individual channel images are displayed side-by-side in one figure.
* This helps visualize how each RGB component contributes to the overall color image.
* The combined figure is saved as sea\_RGB\_channels.jpg.

2.

* A new image is created by reordering the RGB channels to BRG (Blue-Red-Green).
* The purpose is to show how changing the order of color channels alters the image appearance.
* A second figure is used to display the original and reordered image side-by-side.
* This output is saved as sea\_RGB.jpg.

3.

* The image is converted into a format that allows mathematical operations on its pixel values.
* Gamma correction is applied to increase contrast — making light areas lighter and dark areas darker.
* Then, the image is darkened by reducing its brightness to only 20% of the enhanced version.
* The original and the final (enhanced and darkened) images are shown together.
* This section is useful for understanding how gamma and brightness adjustments can enhance an image.
* The result is saved as an image.

4.

* The program explores how reducing the number of grayscale levels affects image quality.
* It performs quantization at different bit depths (2-bit, 4-bit, 6-bit, and 8-bit).
* Lower bit depths mean fewer grayscale shades, which leads to more visible distortion or banding.
* Each version of the quantized image is displayed to compare the visual quality at different levels.
* This helps understand the trade-off between image quality and data size in image compression.
* The output images and the entire visualization are saved.

Ex2:

1a.

* Enhances dark regions by applying a logarithmic curve to the intensity values.
* The log function increases the brightness of darker pixels more than brighter ones.
* A graph of the transformation curve and the resulting image is shown.

1b.

* A custom brightness adjustment using defined intensity ranges:
  + Low values are stretched to become brighter.
  + Mid-range values are compressed.
  + High values are capped at 255.
* This transformation is visualized using a mapping graph and the modified image.

2.

* Flipping Horizontally: The image is mirrored along its vertical axis (left ↔ right).
* 180° Rotation: The image is rotated by 180 degrees clockwise.
* Cropping: A central region of the image (half the width and height) is cropped, focusing on the center part.

3.

* The histogram (brightness distribution) of the original image is calculated and displayed.
* Histogram Equalization is applied to redistribute pixel values evenly, enhancing contrast.
* Before-and-after histograms and images are shown side-by-side.

4.

* CLAHE improves local contrast by limiting the effect of very frequent intensity values (clipping).
* A threshold is set to clip the histogram at 30% of the maximum count.
* A virtual image is generated from the clipped histogram to simulate the transformation.
* A custom histogram equalization is applied using a look-up table (LUT).
* The adjusted image is shown alongside the original.

Ex3:

a.

* Two images of cats (cat\_a and cat\_b) are loaded.
* Both images are displayed in their original color format.
* They are then converted to grayscale to analyze their intensity content without color.
* A single figure is used to show all four versions:
  1. Cat A (Color)
  2. Cat B (Color)
  3. Cat A (Grayscale)
  4. Cat B (Grayscale)

b.

* The two original images are first converted to double precision to allow mathematical operations.
* A difference image is calculated using absolute pixel-wise differences between the two.
  + This creates a new image that shows where and how much the two images differ.
* The difference is shown in two ways:
  1. Color Difference – highlights the differences using RGB values.
  2. Grayscale Difference – converts the color difference to a single-channel image for simpler interpretation.
* To make small differences more visible:
  1. Gamma correction is applied to enhance contrast (makes dark differences more noticeable).
  2. The image is brightened by scaling pixel values.
  3. Any pixel values above the allowed maximum (1) are clipped to avoid display errors.

Ex4:

a.

* A color image (waterfall.jfif) is loaded and displayed.
* The image is then converted to grayscale to simplify histogram analysis.
* A histogram of grayscale intensity levels is created. This shows how many pixels fall into each brightness level from 0 (black) to 255 (white).
* The histogram helps identify if the image is low contrast (i.e., pixel values are clustered in a narrow range).

b.

* The image is split into its Red, Green, and Blue components.
* Each color channel is visualized independently by isolating it and setting the other two channels to black.
* Then, a new image is created by reordering the color channels into a BRG format (Blue, Red, Green), which changes the color appearance significantly.

c.

* The grayscale version of the image is used again for contrast enhancement.
* Histogram equalization is applied:
  + This technique spreads out pixel intensities to cover a broader range.
  + It helps enhance details in both dark and bright regions of the image.
* The script then displays both the equalized image and its updated histogram.
* The new histogram typically appears more balanced or flattened compared to the original.

Ex5:

1.

* A noisy image file (weather.png) is loaded.
* The image is displayed in two versions:
  + Color version (original)
  + Grayscale version (simplified, intensity-only view)
* This helps in analyzing how noise appears in both visual formats.

2.

* A Gaussian blur is applied to the image to reduce noise.
  + The Gaussian filter used has a 5×5 size and a standard deviation (σ) of 1.
  + It smooths out minor variations (like random noise) by averaging neighboring pixel values.
* The result is a denoised version of the image where noise is suppressed but edges may be slightly softened.
* A Butterworth high-pass filter is applied to the image.
  + High-pass filters enhance the edges and fine details by amplifying high-frequency components in the image.
  + The Butterworth filter is known for its smooth transition (non-abrupt cutoff) and is applied here in the frequency domain.
* This step helps restore sharpness lost due to noise reduction.
* Three images are displayed side-by-side:
  + The original noisy image
  + The denoised version (after Gaussian filtering)
  + The sharpened version (after Butterworth high-pass filtering)