Assignment — Week 1 (Beginner, NumPy Only)

Goal:

Learn how images are stored as matrices, how color channels work, and how simple filters affect images — all using **NumPy**, no OpenCV.

Part A — Pixel and Color Basics

Q1. Load and Display an Image

Use `matplotlib.image.imread()` to read any image of your choice.

- Print its shape.
- Check the range of pixel values (min, max).
- Display the image using `matplotlib.pyplot.imshow()`.

Q2. Separate and Combine Channels

1. Extract the **R**, **G**, and **B** channels using NumPy slicing.
   Example:

   ```
   R = img[:, :, 0]
   G = img[:, :, 1]
   B = img[:, :, 2]
   ```

2. Display each channel separately in grayscale.
3. Combine the channels back using `np.stack([R, G, B], axis=-1)` and show the recombined image.

Q3. Convert to Grayscale (Manually)

Use the formula:

Gray=0.299R+0.587G+0.114BGray = 0.299R + 0.587G + 0.114BGray=0.299R+0.587G+0.114B

Display the grayscale image and compare with `np.mean(img, axis=2)` (simple average).

Part B — Understanding Color Spaces

Q4. Create an Artificial RGB Image

Create a small (3×3) image manually as a NumPy array, like:

```
img = np.array([
  [[1,0,0],[0,1,0],[0,0,1]],
  [[1,1,0],[0,1,1],[1,0,1]],
  [[0.5,0.5,0.5],[1,1,1],[0,0,0]]
])
```

Print each pixel's RGB values and describe what color you expect it to be.

Q5. Simple RGB to "Brightness"

Write a function:

```
def brightness(img):
    return np.max(img, axis=2)
```

Display the brightness map and note which areas look bright or dark.

Part C — Filters and Convolution

Q6. 3×3 Blur Filter

Write a function:

```
def blur_gray(img):
    kernel = np.ones((3,3)) / 9
    # use manual convolution (loops)
    ...
```

Apply it on your grayscale image and display the result.
(Hint: pad the borders with zeros using `np.pad()`.)

Part D — Reflection

Answer briefly:

1. What does a convolution kernel do?
2. How does changing the kernel affect the output?
3. Why does grayscale use weighted averages instead of a simple mean?

Submission:

- One `.ipynb` notebook (code + output + small explanations)
- You can use any small image (≤ 512×512)
- Optional: include before/after comparison plots