Python recap(Optional)

1. For a given string find the longest word.
2. Write a function which gets an integer number as input and adds to that number it's reverse, until the result is palindrome. A palindrome number is an integer that reads the same forwards and backwards. In other words, it is a number that remains unchanged when its digits are reversed. The function should return the number of steps.
   Ex. 1) 123+321 = 444 is palindrome, so function returns 1.
   2) 555 is palindrome, so the function returns 0.
   3) 49+94 = 143 isn't palindrome, 143+341 = 484 is palindrome, so function returns 2.
3. Write a function that gets a square matrix as input and returns the sum of the primary diagonal.
4. Write a function that gets a square matrix and swaps the primary and secondary diagonals.
5. Write a function to calculate the sum above primary and secondary diagonals.

P.S. If you find these problems too easy, try problems from leetcode!

1. Binary search
2. Remove duplicates from sorted array
3. Happy number
4. Add digits

Image preprocessing tasks

1.**Install** OpenCV

2. Write a program in Python that:
   Reads an image from a file
   Displays the image in a window
   Waits for any key press to close the window

3. Learn and implement various image manipulations such as **resizing, flipping, cropping, rotating, and validating image loading**.

Detailed Steps:

- **Load** an Image:
  Load an image and **check** if the image is loaded correctly (it is **not empty**).

- **Resize** the Image:
  Change the image size.

- **Flip** the Image:
  Flip the image **vertically and horizontally**.

- **Crop** the Image:
  Crop a specific region from the image.

- **Rotate** the Image:
  Rotate the image by a certain angle.

- Convert to different color spaces:
  Convert the image to **RGB, HSV, LAB and Grayscale** color spaces.

- **Draw shapes**:
  Draw basic shapes: **lines, rectangles, and circles** with specified parameters for **position, dimensions, color, and thickness**.

- Apply image filters:
  Apply Gaussian Blur to reduce image noise and soften details.
  Apply a sharpening filter using a **custom kernel** to enhance edges and details.
  Apply median filtering to effectively remove salt-and-pepper noise.
  (Use appropriate photos to see the changes after applying the filters)

- **Display** and **save** all transformations:
  Display each transformation using OpenCV's display functions.
  Save each transformed image.