The methods you need to edit include `print_fundamental_matrix`, `calculate_alignment`, `align_images` (and if you choose Level 2, `detect_feature_points`).

Problem 0

Method(s) to edit: `calculate_alignment`

Goal: Create two arrays of matched points using `orb_detection_result`.

Problem 1

Method to edit: `print_fundamental_matrix`

Goal: Compute the fundamental matrix using RANSAC.

Problem 2

Method to edit: `calculate_alignment`

Goal: Calculate homography using RANSAC.

Problem 3

Method to edit: `align_images`

Goal: Warp image1 using given `h` and dimensions of `image2`.

Problem 4 (Level 2 only)

Method to edit: `detect_feature_points`

1. Create a new ORB detector.
2. Use the ORB detector to produce two pairs of keypoints and descriptors.
3. Compute the orb_matches parameter (*Hint:* use `cv2.BFMatcher`).

Optional

Create a method named `crop_black` that inputs an image and outputs a modified image that crops off the ”black” parts of the warped image.

Hints The code in function `preprocess_crop_black` returns a list of points that represent the convex hull of the largest contour (which is image itself) inside the warpped image. You can exhaustively enumerate all possible rectangles within the largest contour and find the one with the largest area to completely crop off parts of the image that contain the black parts. There are only $\binom{n}{2}$ which is $O(n^2)$ possible rectangles to try.