Using This Code

To use the code we need to first choose corresponding points between the first and second images. Calling the function `get_coords(img1, img2)` will return these points. The user may specify any number of correspondence points so long as there at least
four. We then compute the homography, `compute_homography(coords)`, and then apply it to the images, `warp(H, img1, img2)`.

Example:

```matlab
img1 = imread('uttower1.JPG');
img2 = imread('uttower2.JPG');
coords = get_coords(img1, img2);
H = compute_homography(coords);
warp1 = warp(H, img1, img2);
imshow(warp1);
```

We can also double check our homography matrix before applying it by using the `verify_points(H, coords, img2)` function. This function will display the points (coords) chosen to create the homography matrix. As a final check, the function `pick_n_apply(H, img1, img2)` will allow the use to pick additional points and have them translated over using `H`.

**Results**

![UT Tower Composition](image)

![Composition of Three images of Petra](image)
Same Composition But Different Ordering When Applying Warp()

A New Masterpiece Has Been Added

Extra Credit

Artwork On The Floor
(courtesy:
The Artwork Rectified