1. Write a function to perform real-valued indexing on an image. Your function should receive an image \( I \) and the (real-valued) pixel coordinates \( \langle r, c \rangle \) and return the RGB values obtained by performing a weighted average of the pixels \( I[\lfloor r \rfloor, \lfloor c \rfloor], I[\lfloor r \rfloor, \lceil c \rceil], I[\lceil r \rceil, \lfloor c \rfloor], I[\lceil r \rceil, \lceil c \rceil] \), where the weights are proportional to the area of overlap of the pixels in the original image and a hypothetical pixel centered at \( \langle r, c \rangle \). Use this function and regular rounding to evaluate the results of each of the methods from the following questions.

2. Write a program to extract a the image of a rectangular object seen under perspective in an image and display it without perspective effects. You can do this using the homography-based algorithm described in the textbook, or the line-based algorithm described in class.

3. Write a program to perform the inverse process from the previous question, that is, it should insert a rectangular region into an image with perspective effects.

4. Write a program to implement k-nearest neighbor warping, as explained in class. Your program should allow the user to input source and destination points and generate a sequence of images illustrating a smooth transition from the source to the destination image.

5. Write a program to morph a face image in frontal view into another. Use a variation of your program from the previous question to align the faces, then apply cross-dissolve.