

LAB 5

*Assigned: February 24-25**Due: February 24-25*

For your fifth lab we'll do a little review along with exploring some new ideas. The emphasis is stepping through pixels based on their location within an image. As always, don't forget that at the end of each chapter contains a Programming Summary section that provides a quick dictionary of all of the important function and encoding names introduced thus far. When you run into problems don't forget to check the Common Bugs and Debugging Tips scattered throughout the chapters as well.

- As always switch the driver (person typing the code) and navigator (person watching, helping spot typos, etc.) on every new problem or every half hour, whichever comes first.
- Do all of the problems in a single python file. Put everyone's name in a comment at the top of the file. Label the start of each problem with a comment.
- For any problem that includes responding to questions as well as interacting with python code, write your responses as a comment along with the required code.

To Do

First lets review looping based on index and notice how the same goal can often be accomplished with if based conditions. While we're at it, we'll practice hierarchical decomposition and code reuse.

1. Program 51 (page 121) converts an image to sepia tones. Let's start by converting this logic that converts a single pixel to sepia tones. In other words, write a separate function that takes in a pixel and changes it to sepia tones. We're going to reuse this logic on the next two problems. (Hint: You just need to extract the per-pixel logic from program 51).
2. Now, write a program in the style of section 4.8 that converts the lower 3/5s of the image into sepia. This program should use a range-based for loop to step through only the pixels that you want to modify. Be sure to make use of your function from problem 1!
3. Now let's redo the previous problem, but use an if statement to make sure we modify pixels only in the bottom 3/5s of the image. This time use a for loop that hands you every pixel (like the loops in the earlier part of chapter 4) and combine it with getX and getY to determine the pixels location. Look at Program 49 (page 117) for an example of this pattern. Be sure to make use of your function from problem 1!
4. For homework you worked on problem 5.3. Now extend that work with your lab partner with some related problems. To start, do 5.4.
5. 5.5
6. 5.6
7. 5.7