CS116 – Lab 4
Due Date: October 3

Purpose: In our last lab we looped through the entire picture and then selected the pixels that we wanted by using a conditional. What if it is possible to loop through only the pixels we wanted in the first place – that would be much more efficient and speed things up. That is what we have going to do here by using a loop inside another loop, called a *nested loop*.

Knowledge: This lab will help you become familiar with the following content knowledge:

- How to use nested loops for accessing a region within a picture

- How to use nested loops when you need to know where the pixels are located in the picture

- How to loop through only part of an array

Task: Before starting this lab, you should have read Chapter 6 in your text. Follow the steps in this lab carefully and complete the assignments.

Assignment 1:

Write a function `flip(picture)` that flips a picture over so that things that were on the left are now on the right. You will want to create a new picture and copy the pixel colors from the original picture to the new picture and return the new picture.
This is NOT the same as the mirror example in the textbook. I want the entire picture flipped and displayed as a single image.

Hint: To accomplish this you will want to create a new resulting picture which initially will be empty. Then use nested loops to access pixels in the entire picture. For each of the pixels in the picture you will need to figure out which pixel from the resulting picture will need to be changed.

Criteria For Success: When you show your original picture and the returned picture you will observe that they are flipped from left to right.

Assignment 2:

Write a function `overlay(picture,background,x,y)` that overlays the picture on top of a background picture with its upper left corner at (x,y). It should return a new picture rather than modifying the given ones.

Hint: To accomplish this you will want to create a resulting picture initially containing the background. Then you will want to use nested loops to access pixels in the entire smaller picture. For each of the pixels you will need to figure out which pixel in the resulting picture will need to be changed.

Criteria For Success: When you show your returned picture you will see that the smaller picture has been overlayed on top of the background picture at the position that you specified. Try a couple of different (x,y) locations to check that you can overlay anywhere on the background.

Assignment 3:

Write a function `triangleCrop(picture,x,y,h)` that crops a triangular region of a picture given the x,y coordinates of the upper left corner in the picture and the height of the cropped region. You should return a new picture rather than modifying the given picture.
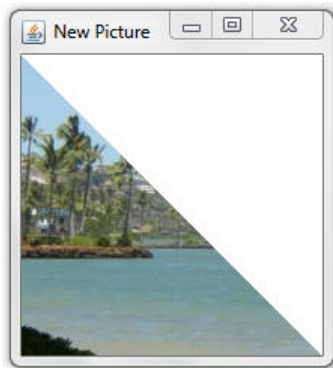
Hint: You will need to modify the nested loops so that the inner loop is a different length each time it is used.

Criteria For Success: Your returned picture should be of a triangle shape. Try a couple of different values for x, y, and h to check that your function works for any *reasonable* values.

Assignment 4:

Write a function `squish(picture)` that squishes a picture vertically (or horizontally).

Hint: Copy every other row (or column) by using the increment in the range.

Criteria For Success: When you show both your original picture and your returned picture you will observe that the new picture is squished.

Assignment 5:

Either in the command window or in a little function, create a collage of at least three images. You should use overlay and any other picture function that we have created or are in the textbook. If you have a picture named p, you can write this picture out to a file with:

`p.write(filepath)`

where `filepath` is a string containing a filepath where you want the file written. A filepath will look something like `"/Users/jzimmerm/Desktop/myCollage.jpg"`

Criteria For Success: You should have a jpeg file that contains your beautiful collage.

Submit your file containing your functions as well as your collage file. Please indicate both partner names in your submission file.