Homework 5

Comp 123

1 Overview

This assignment should be done as individual work. Be sure the work is your own. Do not seek out solutions online. DO ask for help from me or our preceptors. DO write down any people or web sites where you get some measure of help.

For this assignment, I will be taking seriously the expectation that you include a summary, descriptive comment for each function or script you write. I am providing you with a test file for testing the functions you write. Read it carefully.

1.1 Assignment goals

This assignment covers: strings, dictionaries, files, and image manipulation. You have begun to learn a lot of new tools for manipulating images. Bear in mind that what we are learning here is not "standard" Python, but an add-on library, and there are other ways of representing and manipulating images in Python out there.

1.2 Preparing and handing in the assignment

Download the `hw5Code.py` and `hw5Tests.py` files to use with this assignment. Put all your programming answers into the `hw5Code.py` file. Non-programming answers may be put in that file as well.

Be sure that each function you write is preceded by a hash-mark comment that indicates the problem number, and includes inside it a triple-quoted, one-two sentence description of the purpose of the program.

Unless we discuss otherwise in class, complete this assignment using the `imageTools` module in PyCharm.

2 Assignment questions

1. (10 pts) Create a function called `encrypt(filename1, filename2, cipher)` that has three parameters: `filename1` and `filename2`, which are strings representing the path to a text file, and `cipher`, which is a dictionary that has a mapping from each letter of the alphabet (so the keys are 'a', 'b', 'c', etc.) to a different one. This function should:

   - Open up `filename1` for reading
   - Read in `filename1`

- Character by character, encrypt the letters in `filename1` based on the key:value pairs in the `cipher` dictionary (remember that Python is cAse SensITiVE... the dictionary will only have lower case values, so you will have to change all the characters to lower case first, and you will have to ignore all punctuation/numbers)
- Open up `filename2` for writing
- Write the encrypted characters to `filename2`
- Close both the opened files

The function does not return anything.

For instance, if we were using rot13 encoding, where the alphabet is shifted by 13 letters, `cipher` could look something like this: 'a':'n', 'b':'o','c':'p','d':'q' (etc.) and a sentence like "Hello, world" in rot13 encryption would be "uryyb, jbeyq!"

We have provided a decryption algorithm for you to test that your function works correctly.

2. (10 pts) Create a function called `colorSwap` that takes a `Picture` object as its input. Your function should first make a copy of the input `Picture`. Then, it should iterate over the pixels in the new picture, and it should swap the color channels. The old red value should become the new blue, the old green value should become the new red, and the old blue value should become the new green.

   Below is an example of how to test this function, and what its results should be:

```
p1 = makePicture('redDoor.jpg')
show(p1)
p2 = colorSwap(p1)
show(p2)
```

p1

p2

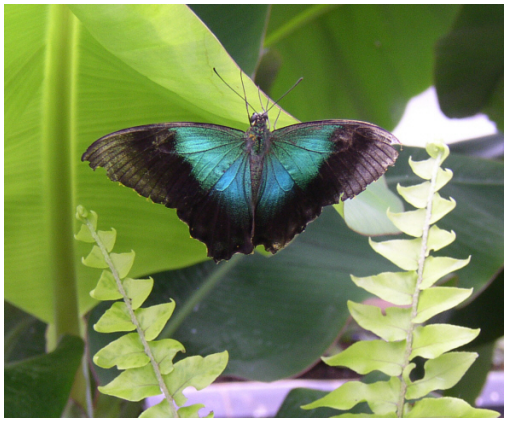
Note: The `hw5Tests.py` file contains a testing function for this, though some parts must be checked by eye.

3. (10 pts) Define a function, `redToCyan` that takes in a Picture object as its input. It should create a new picture that is a copy of the original, and then it should loop over all the pixels in the copy. If the pixel is "red enough," then the pixel's color should be changed to cyan (red is 0, and green and blue are both 255). A pixel is red enough if its red value is greater than both blue and green values.

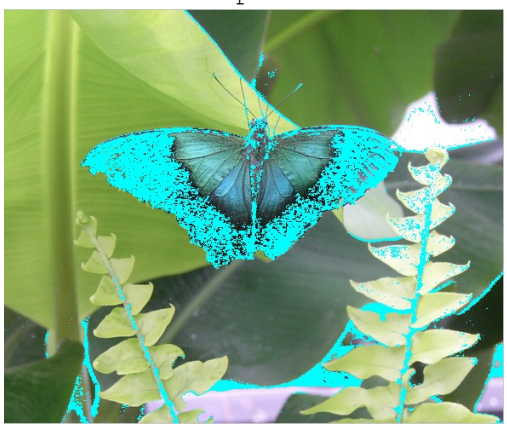
   Below is an example of how to test this function, and what its results should be:

```
p1 = makePicture('butterfly2.jpg')
show(p1)
p2 = redToCyan(p1)
show(p2)
```

p1



p2



   Use the test calls in the `redToCyanTests` function to check if your function is working correctly.