Assignment 3: A library of image-processing functions

In this assignment, you'll create a library of functions in Python that can be used to process images. There are a few important housekeeping items to keep in mind:

1. Each of your functions must conform to the specifications given in this assignment. In other words:
   - The function name must be identical to the name given in this assignment.
   - The number and data type(s) of the function parameters must conform to the specifications.
   - The value that the function returns must be of the data type given in the assignment.

2. Each function must be properly documented.

3. You must write all your functions in one Python file that you will turn in on Monday.

Time frame:

<table>
<thead>
<tr>
<th>Day</th>
<th>Activity</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wednesday</td>
<td>Introduce assignment, review workflow, ask/answer questions, begin work</td>
<td>reduceRed()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reduceGreen()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reduceBlue()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cleared()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>clearGreen()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>clearBlue()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>makeRed()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>makeGreen()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>makeBlue()</td>
</tr>
<tr>
<td>Thursday</td>
<td>Work on part 1. If you don’t get Part 1 done by the end of class, finish it for homework.</td>
<td>flipHorizontal()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>flipVertical()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rotateRight()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rotateLeft()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rotate180()</td>
</tr>
<tr>
<td>Friday</td>
<td>Work on Part 2. If you don’t get Part 2 done by the end of class, finish it for homework.</td>
<td>warhol()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mirror1()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mirror2()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mirror3()</td>
</tr>
<tr>
<td>weekend</td>
<td>Finish writing all functions!</td>
<td></td>
</tr>
<tr>
<td>Monday</td>
<td>Test your functions by doing Part 3 and hand in. Late homework will not be accepted!</td>
<td></td>
</tr>
</tbody>
</table>

Copyright 2007-2008 Marty Billingsley
Intro to Computer Science :: Programming in Python

Part 1: Write and test the following general-purpose functions:

Name: **reduceRed**
Purpose: reduces the amount of red in a picture
Parameters: a Picture
an integer 0-100 that indicates the percentage of red to be left in the picture
Return value: Picture
Side effects: none
Example: myNewPict = reduceRed(myPict, 40)
At the end of this statement, myNewPict is a copy of myPict with only 40% of the red; myPict is unchanged.

Similarly, write and test **reduceGreen** and **reduceBlue**

Name: **clearRed**
Purpose: removes all the red in a picture
Parameters: a Picture
Return value: Picture
Side effects: none
Example: myNewPict = clearRed(myPict)
At the end of this statement, myNewPict is a copy of myPict with no red; myPict is unchanged.

Similarly, write and test **clearGreen** and **clearBlue**

Name: **makeRed**
Purpose: makes a picture look red by removing all the green and blue
Parameters: a Picture
Return value: Picture
Side effects: none
Example: myNewPict = makeRed (myPict)
At the end of this statement, myNewPict is a copy of myPict that looks very red; myPict is unchanged.

Similarly, write and test **makeGreen** and **makeBlue**
Part 2: Write and test the following functions:

Name:      flipHorizontal
Purpose:   flips a picture horizontally
Parameters: a Picture
Return value: Picture
Side effects: none
Example:   myNewPict = flipHorizontal (myPict)
           At the end of this statement, myNewPict is a copy of myPict that
           has been flipped horizontally; myPict is unchanged.

Name:      flipVertical
Purpose:   flips a picture vertically
Parameters: a Picture
Return value: Picture
Side effects: none
Example:   myNewPict = flipVertical (myPict)
           At the end of this statement, myNewPict is a copy of myPict that
           has been flipped vertically; myPict is unchanged.

Name:      rotateRight
Purpose:   rotates a picture 90° clockwise
Parameters: a Picture
Return value: Picture
Side effects: none
Example:   myNewPict = rotateRight (myPict)
           At the end of this statement, myNewPict is a copy of myPict that
           has been rotated clockwise; myPict is unchanged.

Similarly, write and test rotateLeft function that rotates 90° counter-clockwise

Name:      rotate180
Purpose:   rotates a picture 180°
Parameters: a Picture
Return value: Picture
Side effects: none
Example:   myNewPict = rotate180 (myPict)
           At the end of this statement, myNewPict is a copy of myPict that
           has been rotated 180°; myPict is unchanged.
Part 3: Using your functions

Type the following function into your definitions window:

```python
def warhol (pict):
    w = getWidth(pict)
    h = getHeight(pict)
    newPict = makeEmptyPicture(w * 2, h * 2)
    copyInto (pict, newPict, 1, 1)
    copyInto (clearRed(pict), newPict, w + 1, 1)
    copyInto (clearGreen(pict), newPict, 1, h + 1)
    copyInto (clearBlue(pict), newPict, w + 1, h + 1)
    return newPict
```

Test the warhol() function, using a photo of yourself take with PhotoBooth.

Type the following functions into your definitions window:

```python
def mirror1 (pict):
    w = getWidth(pict)
    h = getHeight(pict)
    newPict = makeEmptyPicture(w, h * 2)
    copyInto (pict, newPict, 1, 1)
    copyInto (flipVertical(pict), newPict, 1, h+1)
    return newPict

def mirror2 (pict):
    w = getWidth(pict)
    h = getHeight(pict)
    newPict = makeEmptyPicture(h, w * 2)
    copyInto (rotateRight(pict), newPict, 1, 1)
    copyInto (rotateLeft(pict), newPict, 1, w+1)
    return newPict

def mirror3 (pict):
    w = getWidth(pict)
    h = getHeight(pict)
    newPict = makeEmptyPicture(w, h)
    leftHalf = cropPicture (pict, 1, 1, w/2 + 1, h)
    copyInto (leftHalf, newPict, 1, 1)
    copyInto (flipHorizontal(leftHalf), newPict, w/2 + 1, 1)
    return newPict
```

Test your functions with a non-square picture.

*Be prepared to turn in your Python file.*