Group Exercise 20: Embedding and Manipulating Images
CPSC 301 - 2016W2

NOTE ON SOLUTIONS: As in Group Exercise 19 Q2, there are actually a couple of different ways to solve this problem depending on how you interpret the specification, and whether you prioritize centering the embedded image or having the image be as close to half the width/height in size as possible. So acceptable solutions may differ in their precision by a couple of pixels.

1. Complete the following function embed() according to its docstring, and without using the method paste():

```python
def embed(image1, image2):
    '''
    (RGB-image, RGB-image) -> RGB-image
    Accepts two images already opened for processing and returns a new image that displays the second image in the center of the first image. No matter what the original size of the second image is, the second image's size in the result is half of the size of the first image.
    '''
    (w1, h1) = image1.size

    img = image1.copy()
    img2 = image2.resize((w1//2, h1//2))

    start_i = w1//4
    start_j = h1//4

    for i in range(w1//2):
        for j in range(h1//2):
            img.putpixel((start_i+i, start_j+j), img2.getpixel((i,j)))

    return img
```
2. Complete the following function `embed_hflip()` according to its docstring, and *without* using the method `paste()`. The function will be similar to that in part 1, but with a modification so that the embedded image is also flipped horizontally.

```python
def embed_hflip(image1, image2):
    
    (RGB-image, RGB-image) -> RGB-image
    Accepts two images already opened for processing and returns a new image that contains the second image, flipped horizontally and in the center of the first image. The second image's size in the result is half of the size of the first image.
    
    (w1, h1) = image1.size

    (w1, h1) = image1.size
    img1 = image1.copy()
    img2 = image2.resize((w1//2, h1//2))
    start_i = w1//4
    start_j = 3*h1//4

    for i in range(w1//2):
        for j in range(h1//2):
            img1.putpixel((start_i+i, start_j-j), img2.getpixel((i,j)))

    return img1
```

Some Image functions you might find useful
- `image.size`: returns the size of the image, i.e. a tuple (w, h)
- `image.getpixel((x,y))`: returns the color for the pixel at position (x,y); the color is a triplet (r,g,b); r, g, b are from 0 to 255
- `image.putpixel((x,y), color)`: sets the color of pixel (x,y) to the given color; the color is a triplet (r,g,b); r, g, b are from 0 to 255
- `image.copy()`: returns a new copy of the image
- `image.resize((w,h))`: returns a resized copy of the image. The size is specified by a tuple, where w is the width of the resized image, and h is the height.