

# XOR ROX

Write a Python program that performs bitwise operations (**AND**, **OR**, and **XOR**) on RGB values of each pixel of a PNG image to see the effect that these operations have on the image. Randomly generate a key consisting of random RGB values for each pixel. These values will be used to perform the bitwise operations. For example, suppose that the current pixel of the input image is (10, 122, 221) and the randomly generated key for this pixel is (89, 201, 192), the resulting pixel from a bitwise **AND** operation is (8, 72, 192). Randomly generate new RGB values for each pixel (this is the key) and perform all bitwise operations using the same key.

The input image should be specified in a variable at the top of your program (see below for more details). The output images (i.e., the resulting images from **AND**, **OR**, and **XOR** bitwise operations) should be specified in variables at the top of your program (see below for more details). The key should be output to **stdout**, one comma-separated RGB value per line (see the example below for the required format).

Feel free to use the provided template. It provides the default filenames as follows:

- **INPUT\_IMAGE**: the original PNG image to perform the bitwise operations on (keep the filename as the default so that I can grade these more easily)
- **AND\_IMAGE**: the image resulting from bitwise **AND** on each pixel using the key (keep the filename as the default so that I can grade these more easily)
- **OR\_IMAGE**: the image resulting from bitwise **OR** on each pixel using the key (keep the filename as the default so that I can grade these more easily)
- **XOR\_IMAGE**: the image resulting from bitwise **XOR** on each pixel using the key (keep the filename as the default so that I can grade these more easily)

The template also demonstrates how to load a PNG image, obtain the pixels of the image, modify the pixels, and save the resulting PNG image.

## Notes and Requirements:

- Submit your source code only. I will provide my own input PNG to test with;
- Write the key to **stdout**;
- Output the resulting PNG to files as specified above; and
- Comment your source code appropriately.

Please, no GUIs. Make this a command line application without frills that I can execute at the command line as illustrated below via several sample runs of my program:

```
jgourd@latech:~$ python xor-rox.py
[input.png is loaded]
100,201,201
201,61,240
133,124,56
109,127,110
50,115,93
34,127,31
...
[and.png, or.png, xor.png are all stored]

jgourd@latech:~$ python xor-rox.py > input.key
[input.png is loaded]
[and.png, or.png, xor.png are all stored]
[the key is stored to input.key]
```

Note that I have a program that will take your XOR image and your key, and attempt to recreate the original image.