CYEN 406/CS 444
Sangam Mulmi

Assignment 2: The Magic of XOR

Total Points: 40

Due: April 3, 2017 (Monday) at 11:55 PM

Hey, not-so-Novice Cryptographer! For this assignment, you are required to write a program in **Python 2.7** to perform bitwise operations on RGB values of each pixel on a PNG image file to encrypt the image using the *XOR* operator, and compare it against the results obtained using the *OR* and *AND* operators. The program should take two arguments. **The output image must be written into a new PNG image file** and **the random key used to perform bitwise operation must be written into a text file**, for future decryption purposes. For image input, use a PNG file with NO transparency (NO alpha values.)

Instructions:

- Take **two** arguments from the command-line:
    1. <u>Bitwise operation to perform</u>: **or** or **and** or **xor**.
    2. <u>Name of the input PNG image file</u>: **image_file_name.png**.
- Read RGB values from the input image file, one pixel at a time.
    - <u>Example values</u>: (255, 255, 255) for pixel [0, 0], (254, 0, 201) for pixel [0, 1], and so on.
- Write a function to generate a random value between 0 to 255. Generate one random value at a time for RGB values individually for each of the pixels. The entire list of these random numbers is basically going to be your encryption key. In total, you will be generating:

  *3 * Total Number of Pixels in an Image* random values.
- Write each of the generated random value into an output text file.
- Perform a bitwise operation (Python allows performing bitwise operations on integers directly, which should make your life easier) on RGB values of each of the pixels, based on the type of operation specified through the passed argument.
- Update the original RGB values with the newly obtained values.
- Write the new image containing updated colors into a new image file.
- Use the same image to perform all three types of operations, then observe and document the differences.

Sample Execution Commands:

OR	`#  python bitwise_encryption.py or t-rex.png` *Output:* `encrypted_or.png` and `key_or.txt`
AND	`#  python bitwise_encryption.py and t-rex.png` *Output:* `encrypted_and.png` and `key_and.txt`
XOR	`#  python bitwise_encryption.py xor t-rex.png` *Output:* `encrypted_xor.png` and `key_xor.txt`

Please note that the keys you store for *OR* and *AND* are useless, since you will not be able to decrypt using those keys. I am asking you to store them anyway, just to keep your program uniform for all three operators, so it will be easier for you to code. The key used for *XOR* operation **must** decrypt the output image, encrypted using *XOR*, successfully.

Submission Guidelines:

1. Write comments on your source code file (*bitwise_encryption.py*). Include *author name*, *date*, *description*, list of resources used, and so on.
2. Create a PDF document (*report.pdf*) containing the copy of original PNG image, and the images that were produced after *OR*, *AND*, and *XOR* operations with the key. Write a short paragraph explaining the observations you made.
3. Compress two (*bitwise_encryption.py* and *report.pdf*) files into a ZIP file (*LastName_Assignment_2.zip*.)
4. Upload the ZIP file to **Moodle** by the deadline.