Table of Contents

Clear workspace 1
B1 1
B2 1
B3, B4 2
B5 2
Export and save images to png files 2

Clear workspace

```
clear all;
clc;
```

B1

```
% input image file here
% max number of pixels is 174762, or 418x418 if image is square
image_path = "images\dimorphos.png";
A = imread(image_path);

fprintf("Input image is %s\n", image_path);
fprintf("Image size is %dx%d.\n", size(A, 1), size(A, 2));

% Check if the image is too large to be processed
% If it is, offer the user the choice to resize it, otherwise exit
if (size(A,1) * size(A,2) > 174762)
    prompt = "Image is too large to process, do you wish to resize? Y/n\n";
    x = input(prompt, "s");
    if (x == 'Y') || (x == 'y')
        fprintf("Resizing image to 418x418.\n");
        A = imresize(A, [418, 418]);
    else
        fprintf("Exitting script, please input valid input (less than 174762
 pixels, or 418x418).\n");
        return
    end
end

% Create original image figure
figure('Name', 'Original Image');
clf;
image_original = imshow(A);
image_original_export = gca;
```

B2

```
my_random_numbers; % Call the script to load the random-generated numbers
```

```
RAND_matrix = RANDOM_DATA_OUT(1:numel(A)); % Load random data into array
```

B3, B4

```
% Initialize encrypted image array, and convert to uint8
A_encrypted = zeros(size(A));
A_encrypted = uint8(A_encrypted);

for i = 1:numel(A_encrypted)
    % XOR the original image with the random data to encrypt it
    A_encrypted(i) = bitxor(A(i), RAND_matrix(i));
end

% Create encrypted image figure
figure('Name', 'Encrypted Image');
clf;
image_encrypted = imshow(A_encrypted);
image_encrypted_export = gca;
```

B5

```
% Initialize encrypted image array, and convert to uint8
A_decrypted = zeros(size(A_encrypted));
A_decrypted = uint8(A_decrypted);

for i = 1:numel(A_decrypted)
    % XOR the encrypted image with the random data to decrypt it
    A_decrypted(i) = bitxor(A_encrypted(i), RAND_matrix(i));
end

% Create decrypted image figure
figure('Name', 'Decrypted Image');
clf;
image_decrypted = imshow(A_decrypted);
image_decrypted_export = gca;
```

Export and save images to png files

```
exportgraphics(image_original_export,'images/original_image.png');
exportgraphics(image_encrypted_export,'images/encrpyted_image.png');
exportgraphics(image_decrypted_export,'images/decrypted_image.png');
```

Published with MATLAB® R2022b