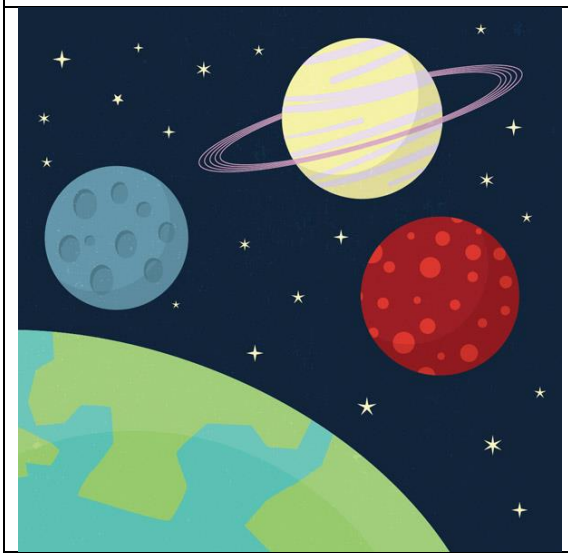
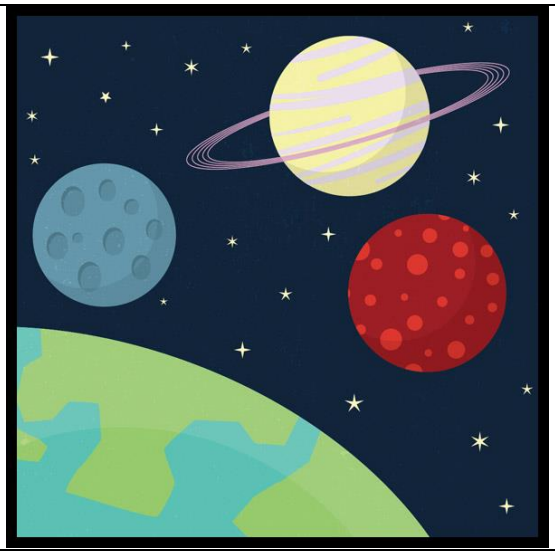
Python Assignment 4 - Images

1. Add Borders

```
addBorders(image, thickness, colour)
```

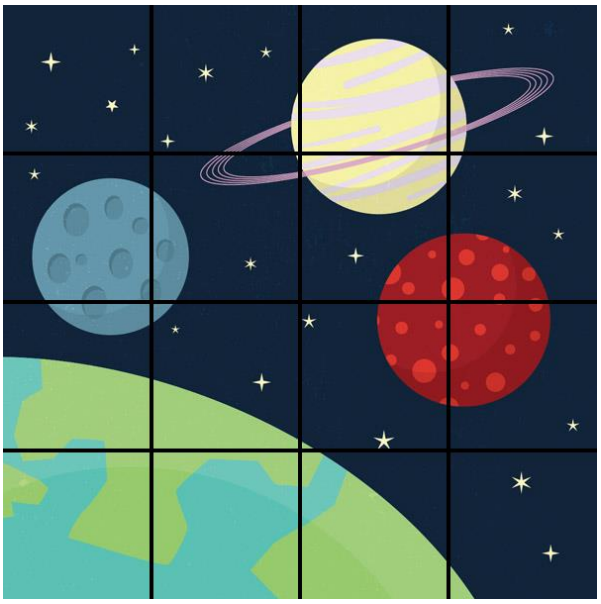
Before	After
	

2. Add Dividers

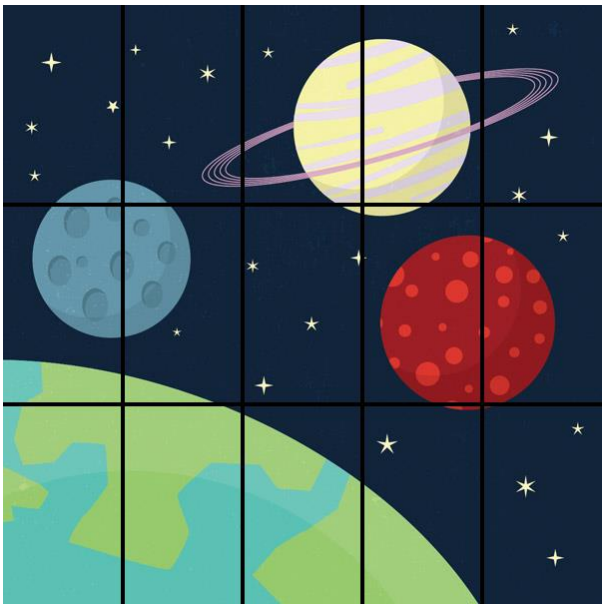
```
addDividers(image, rows, cols, thickness, colour)
```

Might be easier to get it working with rows = cols, then get it to work more generally

Sample Result - 4x4



Sample Result - 3x5



3. Create Image from Binary

```
createImageFromBinary(sourceFileName, targetFileName)
```

Background

Sample images are stored in two text files - "pixels_small.txt" and "pixels_big.txt". Each file contains one string, and that string is all 1's and 0's. Each set of 32 bits represents one pixel. (You can open the small one in a text editor, but the big one is very, very big and may not load.

Structure of the Binary Data

- The first 32 bits is the width of the image to be constructed. (This is metadata, not part of the image itself.)
- The second 32 bits is the height of the image. (Also metadata)
- All remaining bits, in sets of 32, are pixels. For each set of 32 bits:
    - The first 8 bits are the alpha. You should ignore this. (All of our images are opaque.)
    - The second 8 bits are the red value.
    - The third 8 bits are the green value.
    - The last 8 bits are the blue value.

Example Pixel

The string "11111111110110101010000000100000" is an orangey-yellow pixel. Here's why:

- Alpha: 11111111 (255)
- Red: 11011010 (218)
- Green: 10100000 (160)
- Blue: 00100000 (32 – note the leading 0's)

Expected Output Image from "pixels_small.txt"

Expected Output Image from "pixels_big.txt"

(actual size is much bigger)

4. Save Image as Binary

```
saveImageAsBinary(sourceFileName, targetFileName)
```

Create a text file containing a single string. That string contains the binary encoding of an image, in the same format as the files "pixels_small.txt" and "pixels_big.txt". (Make sure you include the image's width and height in the first 64 bits.)

You can use any images you like, but "teapot.jpeg" and "road.jpeg" are included as samples of small and big images, respectively.