Practice S06P07: Image Processing I: Flipping
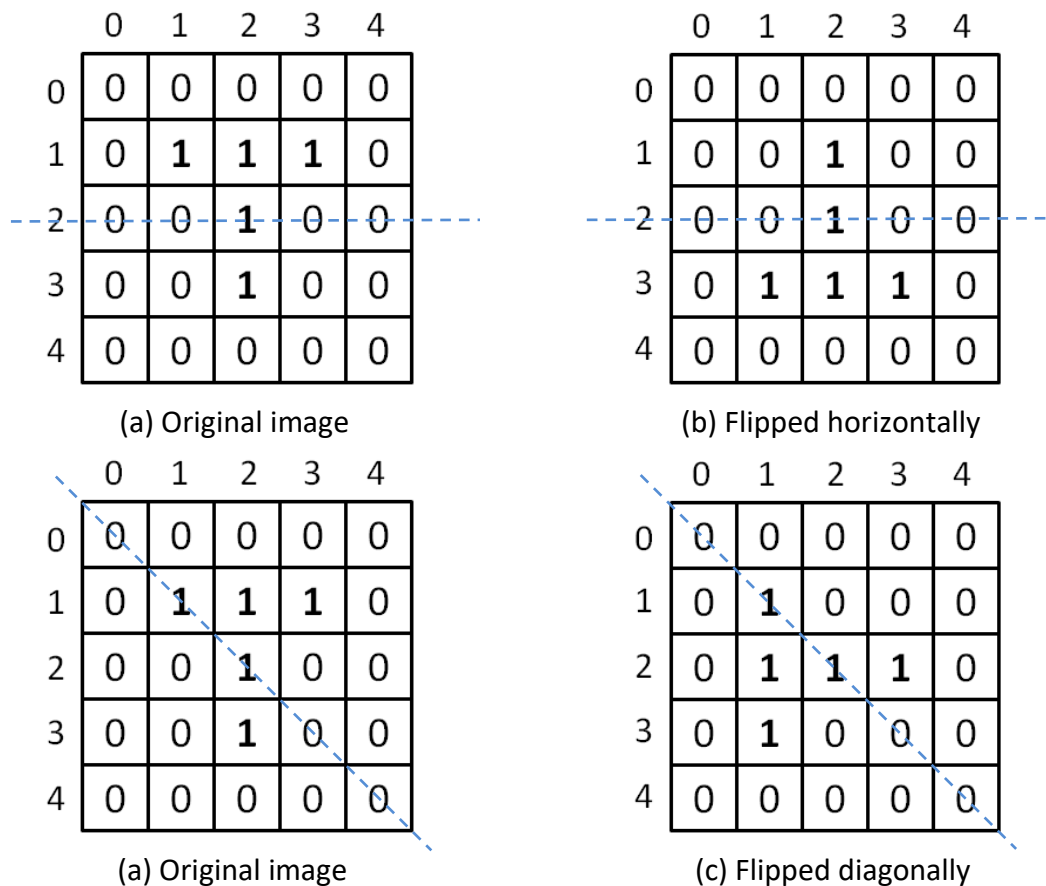
http://www.comp.nus.edu.sg/~cs1010/4_misc/practice.html

Week of release: Week 6

Objectives: 2D array

Task statement:

A square black and white image can be represented as an N x N array of 1s (black pixels) and 0s (white pixels). An image can be flipped in various ways, such as horizontally, vertically, along the diagonal and along the anti-diagonal. Figure (a) shows a sample image where N = 5, while (b) and (c) show the resulting image of flipping the sample image vertically and diagonally, respectively.

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	0
2	0	0	1	0	0
3	0	0	1	0	0
4	0	0	0	0	0

(a) Original image

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	1	0	0
2	0	0	1	0	0
3	0	1	1	1	0
4	0	0	0	0	0

(b) Flipped horizontally

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	0
2	0	0	1	0	0
3	0	0	1	0	0
4	0	0	0	0	0

(a) Original image

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	0	0	0
2	0	1	1	1	0
3	0	1	0	0	0
4	0	0	0	0	0

(c) Flipped diagonally

Write a program **flip.c** to perform the following:

- Read in a positive integer value *size*, which indicates the number of rows and columns of the image. You may assume that *size* is at most 10.

- Read in *size* * *size* integers (0 or 1), which represent the image.

- Read in a character ('h' or 'd'), which represent the direction of flipping.
- If the character is 'h', flip the image horizontally; if the character is 'd', flip the image diagonally.

Your program should have a function called **scan()** to read in the size of an image as well as the actual image, a function called **flipH()** to flip the image horizontally, a function **flipD()** to flip the image diagonally. A function called **print()** is given for printing an image.

Sample run #1:

```
Enter size: 3
Enter values:
1 1 1
0 0 1
0 0 0
Enter direction: h
The image after flipping:
0 0 0
0 0 1
1 1 1
```

Sample run #2:

```
Enter size: 3
Enter values:
1 1 1
0 0 1
0 0 0
Enter direction: d
The image after flipping:
1 0 0
1 0 0
1 1 0
```