

Practice S06P08: Image Processing II: Rotation

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 $N \times N$ array of 1s (black pixels) and 0s (white pixels). Given a degree k (which is a non-negative multiple of 90), an image can be rotated clockwise by k degrees. Figure (a) shows a sample image where $N = 5$, while (b), (c) and (d) show the resulting image of rotating the sample image 90, 180 and 270 degrees, 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	0	1	0
2	0	1	1	1	0
3	0	0	0	1	0
4	0	0	0	0	0

(b) Rotated 90 degrees clockwise

	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

(c) Rotated 180 degrees clockwise

	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

(d) Rotated 270 degrees clockwise

Write a program **rotate.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 an integer k (a nonnegative multiple of 90), which is the degree of the rotation.
- Rotate the image clockwise by k degrees.

Your program should have a function called **scan()** to read in the size of an image as well as the actual image, and a function called **rotate()** to rotate the image. 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 degree of rotation: 90
The image after rotation:
0 0 1
0 0 1
0 1 1
```

Sample run #2:

```
Enter size: 3
Enter values:
1 1 1
0 0 1
0 0 0
Enter degree of rotation: 180
The image after rotation:
0 0 0
1 0 0
1 1 1
```

Sample run #3:

```
Enter size: 3
Enter values:
1 1 1
0 0 1
0 0 0
Enter degree of rotation: 270
The image after rotation:
1 1 0
1 0 0
1 0 0
```