National University of Singapore
School of Computing
SWS3012: SICP
July 2023

R9A
Arrays and Loops

Problem:

We represent a ***matrix*** as a "2D array" of numbers (which is actually an array of arrays of numbers in Source). For example, the 2×2 matrix $\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$ is represented as `[[1, 2], [3, 4]]`.

Write a function `rotate_matrix` that takes as argument a 2D array that represents a $n \times n$ matrix, and rotates the matrix 90 degrees clockwise.

```
function rotate_matrix(M) {
    // ???
}
```

For example, given the following 4×4 matrix:

```
[[ 1,  2,  3,  4],
 [ 5,  6,  7,  8],
 [ 9, 10, 11, 12],
 [13, 14, 15, 16]]
```

It will be modified to become the following matrix after a 90-degree clockwise rotation:

```
[[13,  9,  5,  1],
 [14, 10,  6,  2],
 [15, 11,  7,  3],
 [16, 12,  8,  4]]
```

The **challenge** is that the matrix rotation must be performed **in-place**. This means that, besides the arrays that are used to store the original matrix, **no additional array or list** can be used for the rotation. The rotated result must be stored in the same space as the original matrix. (**Hint:** use only swaps to perform the rotation.)