**Ilkin Suleymanov**

The idea is simple. Transform each row of source matrix into required column of final image. We will use an auxiliary buffer to transform the image. From the above picture, we can observe that

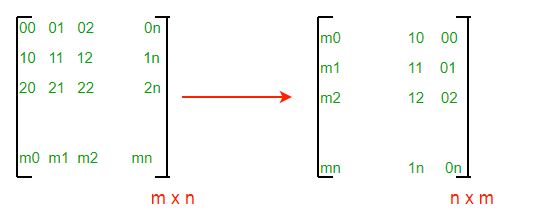
first row of source ------> last column of destination

second row of source ------> last but-one column of destination

so ... on

last row of source ------> first column of destination

In pictorial form, we can represent the above transformations of an (m x n) matrix into (n x m) matrix,



If you have not attempted, at least try your pseudo code now. It will be easy to write our pseudo code. In C/C++ we will usually traverse matrix on row major order. Each row is transformed into different column of final image. We need to construct columns of final image.

Note that there are various ways to implement the algorithm based on traversal of matrix, row major or column major order. We have two matrices and two ways (row and column major) to traverse each matrix. Hence, there can at least be 4 different ways of transformation of source matrix into final matrix.

Result :

1 2 3 4

5 6 7 8

9 10 11 12

9 5 1

10 6 2

11 7 3

12 8 4

**Time Complexity:**O(N\*M), as we are using nested loops for traversing the matrix.

**Auxiliary Space:**O(N\*M), as we are using extra space for matrix.