Sorting Cabs!

Given a vector of pairs representing coordinates x and y of different cabs on a X-Y plane. Sort them according to their distance from the origin in the non-decreasing order i.e. the cab with shortest distance from the origin will be at first.

Input Format

In the function an integer vector of pairs is passed.

Output Format

Return an integer vector of pairs in sorted order.

Sample Input

{ (2,3), (1,2), (3,4), (2,4), (1,4) }

Sample Output

{ (1, 2), (2, 3), (1, 4), (2, 4), (3, 4) }

Solution: sortCabs.cpp

Sort Fruits!

Given a vector of pairs of a string and an integer representing names and prices of different fruits. A string is also passed as a parameter with name equals either "price", meaning you have to sort the fruits on the basis of their price or "name", which means you have to sort the fruit on the basis of their name.

Input Format

In the function an integer vector of pairs is passed.

Output Format

Return an integer vector of pairs in sorted order.

Sample Input

{ (Mango,100), (Guava,70), (Grapes,40), (Apple,60), (Banana,30) }, S = "price"

Sample Output

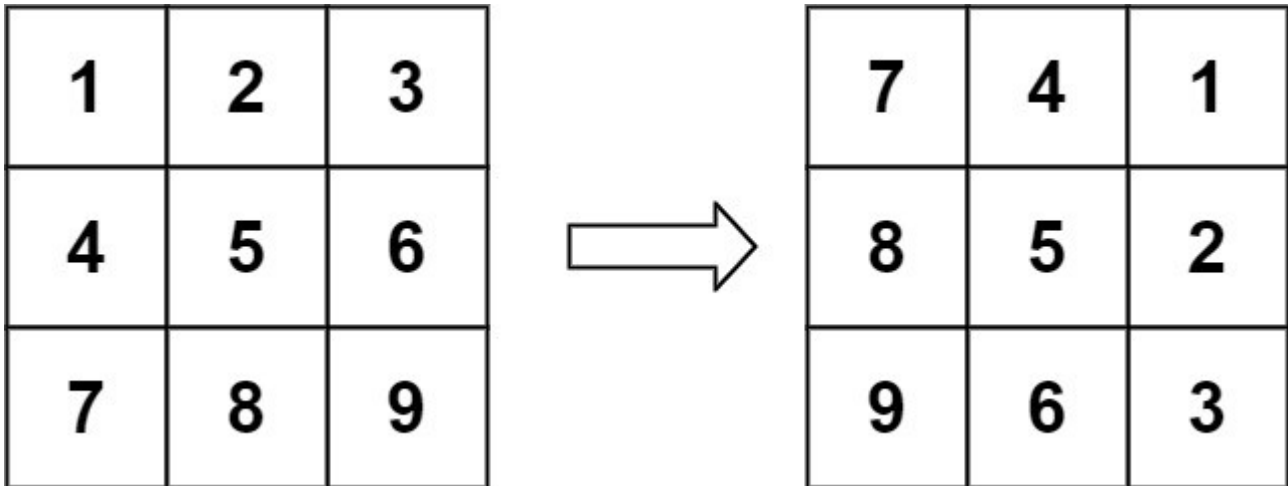{ ("Banana", 30), ("Grapes", 40), ("Apple", 60), ("Guava", 70), ("Mango", 100) }

Solution: sortFruits.cpp & sortFruitsCpp17.cpp

Rotate Image

You are given an n x n 2D matrix representing an image, rotate the image by 90 degrees (clockwise).

You have to rotate the image in-place, which means you have to modify the input 2D matrix directly. DO NOT allocate another 2D matrix and do the rotation.

Example :

Sample Input

```
matrix = [[1,2,3],[4,5,6],[7,8,9]]
```

Sample Output

```
[[7,4,1],[8,5,2],[9,6,3]]
```

Constraints:

- matrix.length == n
- matrix[i].length == n
- 1 <= n <= 20
- -1000 <= matrix[i][j] <= 1000

Solution: imageRotation.cpp

<u>Make Zeroes</u>

Given a two-dimensional array, if any element within is zero, make its whole row and column zero. For example, consider the matrix below

5	4	3	9
2	0	7	6
1	3	4	0
9	8	3	4

Input Format

In the function a 2d vector is passed.

Output Format

Return the updated 2d vector.

Sample Input

{ {5, 4, 3, 9}, {2, 0, 7, 6}, {1, 3, 4, 0}, {9, 8, 3, 4} }

Sample Output

{ {5, 0, 3, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {9, 0, 3, 0} }

Solution: zeroes.cpp