**Print unique rows in a given boolean matrix**

A variant of Trie data structure can be used where each node will be having two children one for 0 and other for 1.

Insert each row in the Trie.

If the row is already there, don’t print the row.

If row is not there in Trie, insert it in Trie and print it.

// A Trie node

typedef struct Node

{

    bool isEndOfCol;

    struct Node \*child[2]; // Only two children needed for 0 and 1

} Node;

**Create a matrix with alternating rectangles of O and X**

Input: m = 3, n = 3

Output: Following matrix

X X X

X 0 X

X X X

Input: m = 4, n = 5

Output: Following matrix

X X X X X

X 0 0 0 X

X 0 0 0 X

X X X X X

Use the logic for printing the elements of matrix in spiral way.

* First cover outer border with X (same as printing outer border in spiral form)
* For next iteration change the character to O
* Again go to next inner circle but flip the character to X

**Count Negative Numbers in a Column-Wise and Row-Wise Sorted Matrix**

Input: M = [-3, -2, -1, 1]

[-2, 2, 3, 4]

[4, 5, 7, 8]

Output : 4

We have 4 negative numbers in this matrix

Here's the idea:

[-3, -2, ↓, ←] -> Found 3 negative numbers in this row

[ ↓, ←, ←, 4] -> Found 1 negative number in this row

[ ←, 5, 7, 8] -> No negative numbers in this row

O(m+n)

**Rotate Matrix Elements**

Given a matrix, clockwise rotate elements in it.

**Examples:**

For 4\*4 matrix

Input:

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

Output:

5 1 2 3

9 10 6 4

13 11 7 8

14 15 16 12

Use loops similar to the [program for printing a matrix in spiral form](http://www.geeksforgeeks.org/print-a-given-matrix-in-spiral-form/).

One by one rotate all rings of elements, starting from the outermost.