Search in a Row-Column sorted matrix

Difficulty: **Easy**   Accuracy: **41.62%**   Submissions: **165K+**   Points: **2**

Given a 2D integer matrix **mat[][]** of size **n x m**, where every row and column is sorted in increasing order and a number **x**, the task is to find whether element **x** is present in the matrix.

Examples:

Input: mat[][] = [[3, 30, 38],[20, 52, 54],[35, 60, 69]], x = 62

Output: false

Explanation: 62 is not present in the matrix, so output is false.

Input: mat[][] = [[18, 21, 27],[38, 55, 67]], x = 55

Output: true
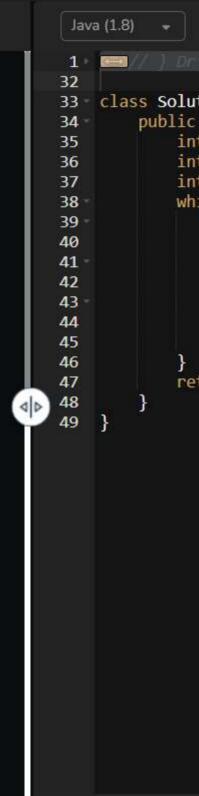
Explanation: 55 is present in the matrix.

Input: mat[][] = [[1, 2, 3],[4, 5, 6],[7, 8, 9]], x = 3

Output: true

Explanation: 3 is present in the matrix.

```java
                 // ) Driver Code Ends

class Solution {
    public static boolean matSearch(int mat[][], int x) {
        int n = mat.length;
        int m = mat[0].length;
        int i = 0, index = m - 1;
        while (i < n && index >= 0) {
            if (mat[i][index] == x) {
                return true;
            } else if (mat[i][index] > x) {
                index--;
            } else {
                i++;
            }
        }
        return false;
    }
}
```

Custom Input   Compile & Run   Submit