ASSIGNMENT - 3

ANIKEIT SETHI(190001003)

**Algorithm:** -

* 1. Let the given element be x, create two variable *i = 0, j = n-1* as index of row and column
  2. Run a loop until i = 0
  3. Check if the current element is greater than x then decrease the count of j. Exclude the current column.
  4. Check if the current element is less than x then increase the count of i. Exclude the current row.
  5. If the element is equal then print the position and end.

**Approach:** The simple idea is to remove a row or column in each comparison until an element is found. Start searching from the top-right corner of the matrix. There are three possible cases.

* 1. **The given number is greater than the current number:** This will ensure, that all the elements in the current row are smaller than the given number as the pointer is already at the right-most element and the row is sorted. Thus, the entire row gets eliminated and continue the search on the next row. Here elimination means that row needs not to be searched.
  2. **The given number is smaller than the current number:** This will ensure, that all the elements in the current column are greater than the given number. Thus, the entire column gets eliminated and continue the search on the previous column i.e. the column at the immediate left.
  3. **The given number is equal to the current number:** This will end the search.

Code:-

# in row-wise and column-wise sorted matrix  
  
# Searches the element x in mat[][]. If the  
# element is found, then prints its position  
# and returns true, otherwise prints "not found"  
# and returns false  
def search(mat, n, x):  
 i = 0  
  
 # set indexes for top right element  
 j = n - 1  
 while i < n and j >= 0:  
  
 if mat[i][j] == x:  
 print("n Found at ", i, ", ", j)  
 return 1  
  
 if mat[i][j] > x:  
 j -= 1  
  
 # if mat[i][j] < x  
 else:  
 i += 1  
  
 print("Element not found")  
 return 0 # if (i == n || j == -1 )  
  
  
# Driver Code  
mat = [[10, 20, 30, 40],  
 [15, 25, 35, 45],  
 [27, 29, 37, 48],  
 [32, 33, 39, 50]]  
search(mat, 4, 29)

Output:-

