Strivers-A2Z-DSA-Sheet-main\02.Binary Search\2D Arrays\2.Search_in_sorted_matrix.cpp

```cpp
/*
QUESTION:-
You are given an m x n integer matrix matrix with the following two properties:

Each row is sorted in non-decreasing order.
The first integer of each row is greater than the last integer of the previous row.
Given an integer target, return true if target is in matrix or false otherwise.

You must write a solution in O(log(m * n)) time complexity.



Example 1:
Input:  matrix = [[1,3,5,7]
                 [10,11,16,20]
                 [23,30,34,60]]
        target = 3
Output: true
*/

/*
APPROACH:-
-> Since the array is sorted we can use binary search low = 0 and high = n*m-1 i.e. total number of elements
-> Value at mid position could be accessed by matrix[mid/m][mid%m]
-> Then, follow the traditional binary search
*/

// CODE:-
bool searchMatrix(vector<vector<int>> &matrix, int target)
{
    int n = matrix.size();
    int m = matrix[0].size();
    int low = 0;
    int high = n * m - 1;
    while (low <= high)
    {
        int mid = low + (high - low) / 2;
        int val = matrix[mid / m][mid % m];
        if (val == target)
            return true;
        else if (val > target)
            high = mid - 1;
        else
            low = mid + 1;
    }
    return false;
}

// TIME COMPLEXITY = O(log(M * N))
// SPACE COMPLEXITY = O(0)
```