int i=(mid/col); // "divide" by "col" to get row value

int j=(mid%col); // "modulo" by "col" to get col value

class Solution {

public:

bool searchMatrix(vector<vector<int>>& matrix, int target) {

int row=matrix.size();

int col=matrix[0].size();

int low=0;

int high=(row\*col)-1;

while(low<=high){

int mid=low+((high-low)/2);

//as if row=3, col=4 ---> then total element is 3\*4=12, so 0 to 11

// int low = 0, high = (r\*c)-1;

// The real thing that you'll get stuck in is how to do the comparison between the mid element and the target element. Its simple, you can find the row index by dividing the mid (index of the number) by number of columns, the quotient will be the row index, while the remainder will be the column index. In simple words, it'll be like:

// matrix[mid/c][mid%c]

// Calculating Row & Col for the Corresponding "mid" value

int i=(mid/col); // "divide" by "col" to get row value

int j=(mid%col); // "modulo" by "col" to get col value

if(matrix[i][j]==target){

return true;

}

else if(matrix[i][j]<target){

low=mid+1;

}

else if(matrix[i][j]>target){

high=mid-1;

}

}

return false;

}

};