class Solution

{

public void dfs(int i,int j,int n,int m, char[][] grid){

// base condition

if(i<0 || j<0 || i==n || j==m || grid[i][j]=='O'){

return;

}

// make the 'X' now 'O'

// It will be visited then

grid[i][j] = 'O';

// now call the dfs for all direction

// upwards direction

dfs(i-1,j,n,m,grid);

// downwards direction

dfs(i+1,j,n,m,grid);

// left direction

dfs(i,j-1,n,m,grid);

// right direction

dfs(i,j+1,n,m,grid);

}

//Function to find the number of 'X' total shapes.

public int xShape(char[][] grid)

{

int n = grid.length;

int m = grid[0].length;

int count=0;

for(int i=0;i<n;i++){

for(int j=0;j<m;j++){

if(grid[i][j]=='X'){

count++;

dfs(i,j,n,m,grid);

}

}

}

return count;

}

}