PROBLEM

Exit Point in a Matrix 🔖

Easy   Accuracy: 50.0%   Submissions: 15K+   Points: 2

Matrix Partners India: Exclusive Job-A-Thon | Apply to 15+ Companies via 1 Hiring Challenge | Starting from 29th April onwards 🔗

Given a **matrix** of size **n x m** with **0's** and **1's**, you enter the matrix at cell **(0,0)** in **left to right** direction. Whenever you encounter a **0** you retain it in the **same direction**, else if you encounter a **1** you have to **change** the direction to the **right** of the current direction and **change that 1** value to **0**, you have to find out from which index you will **leave** the matrix at the end.

Example 1:

```
Input:
n = 3, m = 3
matrix = {{0, 1, 0},
          {0, 1, 1},
          {0, 0, 0}}
Output:
{1, 0}
Explanation:
Enter the matrix at (0, 0)
```

```
Explanation:
Enter the matrix at (0, 0)
-> then move towards (0, 1) ->  1 is encountered
-> turn right towards (1, 1)  -> again 1 is encountered
-> turn right again towards (1, 0)
-> now, the boundary of matrix will be crossed ->hence, exit point reached at 1, 0..
```

Example 2:

```
Input:
n = 1, m = 2
matrix = {{0, 0}}
Output:
{0, 1}
Explanation:
Enter the matrix at cell (0, 0).
Since the cell contains a 0, we continue moving in the same direction.
We reach cell (0, 1), which also contains a 0. So, we continue moving in the same direction, we exit
the matrix from cell (0, 1).
```

Your Task:

You don't need to read or print anything. Your task is to complete the function **FindExitPoint()** which takes the **matrix** as an input parameter and returns a **list** containing the **exit point.**

Expected Time Complexity: O(n * m) where n = number of rows and m = number of columns.
Expected Space Complexity: O(1)

Constraints:
1 <= n, m <= 100

CODE

```python
#User function Template for python3

class Solution:
        def FindExitPoint(self, n, m, matrix):
            i,j,d=0,0,0
    while (0<=i<n and 0<=j<m):
       if matrix[i][j]==1:
          matrix[i][j]=0
          d=(d+1)%4
        if d==0: #Move downwards
          j+=1
        elif d==1: #Move forward towards right
          i+=1
        elif d==2: # Move upwards
          j-=1
        else: #Move forward towards left
          i-=1
    if d==0: return (i,j-1)
    elif d==1: return (i-1,j)
    elif d==2: return (i,j+1)
    return (i+1,j)
                # Code here
```