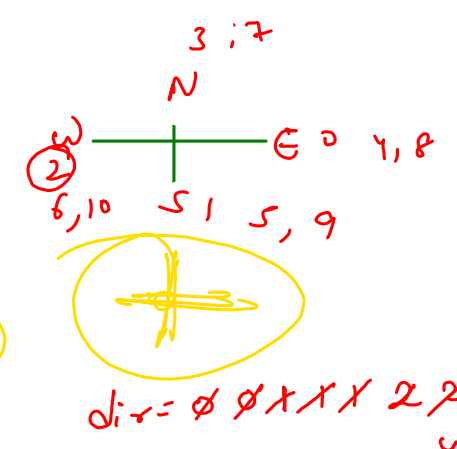


Exit Point -
 Rotate by 90°
 Ring Rotate -
 Saddle Point

Exit Point -

0,0	0,1	1,2	0,3
1,0	1,1	1,2	0,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3



Binary Matrix - 0,1
 Approach:
 4 directions [0, 1, 2, 3]
 $i, j-1 \leftarrow i, j \rightarrow i, j+1$
 $i-1, j \uparrow i, j \downarrow i+1, j$

if (d > 3)

10 % 4 = 2
 4 > 4 = 0
 3 % 4 = 3

```
int dir = 0;
//initially top left 0,0
int i = 0;
int j = 0;
int n = arr.length;
int m = arr[0].length;

while(true){
    dir += arr[i][j];
    dir %= 4;
    if(dir == 0){
        j++;
    }
    else if(dir == 1){
        i++;
    }
    else if(dir == 2){
        j--;
    }
    else{
        //dir == 3
        i--;
    }
    if(i < 0 || j < 0 || i >= n || j >= m){
        break;
    }
    if(i < 0){
        i++;
    }
    else if(i >= n){
        i--;
    }
    else if(j < 0){
        j++;
    }
    else if(j >= m){
        j--;
    }
    System.out.println(i);
    System.out.println(j);
}
```

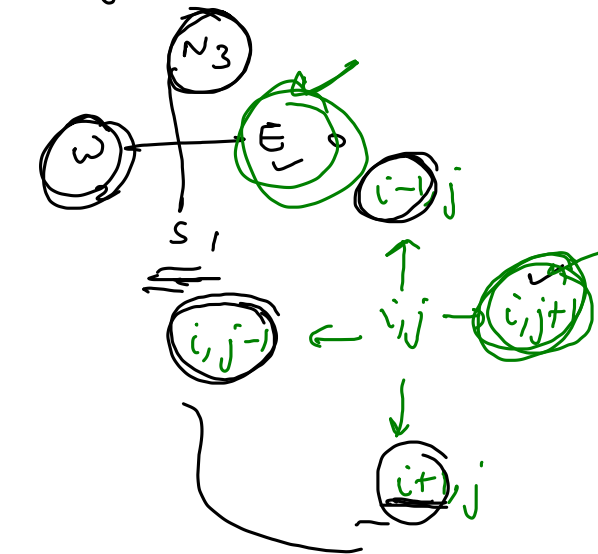
0,0	0,1	1,2	0,3
1,0	1,1	1,2	0,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

i=1
 j=2
 d = 0 % 4 = 0
 1 % 4 = 1



i, j
 1, 2

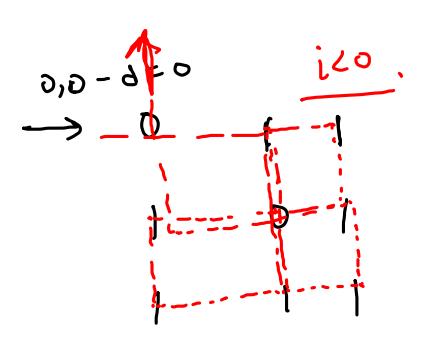
1 → Right 90°



d = 1 % 4 = 1
 1 % 4 = 1
 2 % 4 = 2
 3 % 4 = 3
 4 % 4 = 0

0,0 → 0
 new d = +1

according d
 if (i, j) exit
 i < n
 j < m



4,3
 d = 0
 i = 0
 j = 0

-1,0
 i < 0
 i++
 0,0