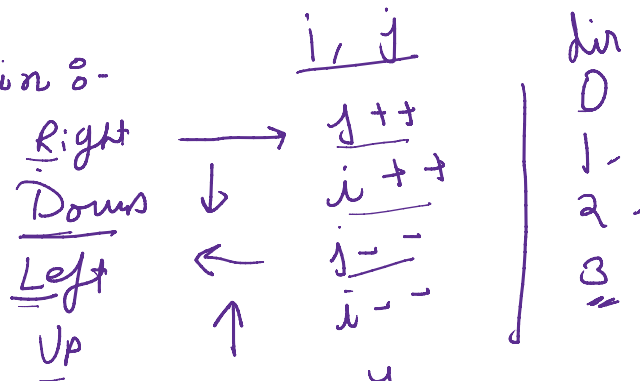


Exit Point of Matrix :-

0 1 2
0 1 0
1 0 1
2 0 0 0
1 0



0 1 0 0
1 0 0 0
0 0 0 0
1 1 0 0

dis = 0
j = 0 x 0 x 2
i = 0 x 2 x 2
j = 4, i = 1
(mat[i][j] + dis) % 4 = 2 x 2 = 4 x 0

4 3
0 0 1
0 0 0
0 0 0
0 0 1

i = 0 x 2 x 2
j = 2 x 0 x 2
dis = 2 x 2 x 2
cd = 2 x 2 x 2

```
int i = 0;
int j = 0;
int dir = 0;
while (i >= 0 && i < mat.length && j >= 0 && j < mat[0].length) {
    int currDir = (dir + mat[i][j]) % 4;
    if (currDir != dir) {
        dir = currDir;
        mat[i][j] = 0;
    }
    if (dir == 0) j++;
    else if (dir == 1) i++;
    else if (dir == 2) j--;
    else if (dir == 3) i--;
}
if (dir == 0) j--;
else if (dir == 1) i--;
else if (dir == 2) j++;
else if (dir == 3) i++;
return new int[]{i, j};
```

[2, 0]

Recursion :-

me() {

me()



100000

3	2	1	A=0
3	2	1	A=1
3	2	1	A=2
3	2	1	A=3
3	2	1	A=4
3	2	1	A=5

```
public static void me(int n) {
    1 if(n==0) return;
    2 System.out.println("Krish");
    3 me(n: n - 1);
}
```

KRISH

KRISH

KRISH

KRISH

KRISH

Print Decreasing No. :-

A=0
A=1
A=2
A=3
A=4
A=5

Print Decreasing No. :-

```
public static void printDecreasingNumber(int n) {
    1 if (n == 0) return;
    2 System.out.println(n);
    3 printDecreasingNumber(n: n - 1);
}
```

pre-order

Print Increasing No. :-

```
public static void printIncreasingNumber(int n) {
    1 if (n == 0) return;
    2 printIncreasingNumber(n: n - 1);
    3 System.out.println(n);
}
```

post-order

5
4
3
2
1

↑

3	2	1	A=0	3	2	1
3	2	1	A=1	3	2	1
3	2	1	A=2	3	2	1
3	2	1	A=3	3	2	1
3	2	1	A=4	3	2	1
3	2	1	A=5	3	2	1

1
2
3
4
5

↓

$A=4$
 $A=5$

$\overline{1002}$
 1000

$A=3$

~~$A=0$~~
 ~~$A=10$~~
 ~~$A=71$~~
 ~~$A=32$~~

x
 $x \ 2 \ 3 \ x$
 $x \ 2 \ 3 \ x$
 $+ \ 2 \ 3 \ x$

```

public static void printIncDec(int n) {
    1  if (n == 0) return;
    2  System.out.println(n + " pre order");
    3  printIncDec(--n);
    4  System.out.println(n + " post order");
}

```

3
 2
 \downarrow
 0
 \downarrow
 2