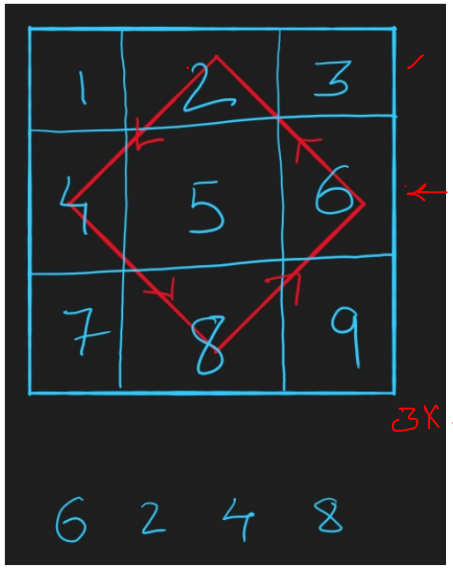
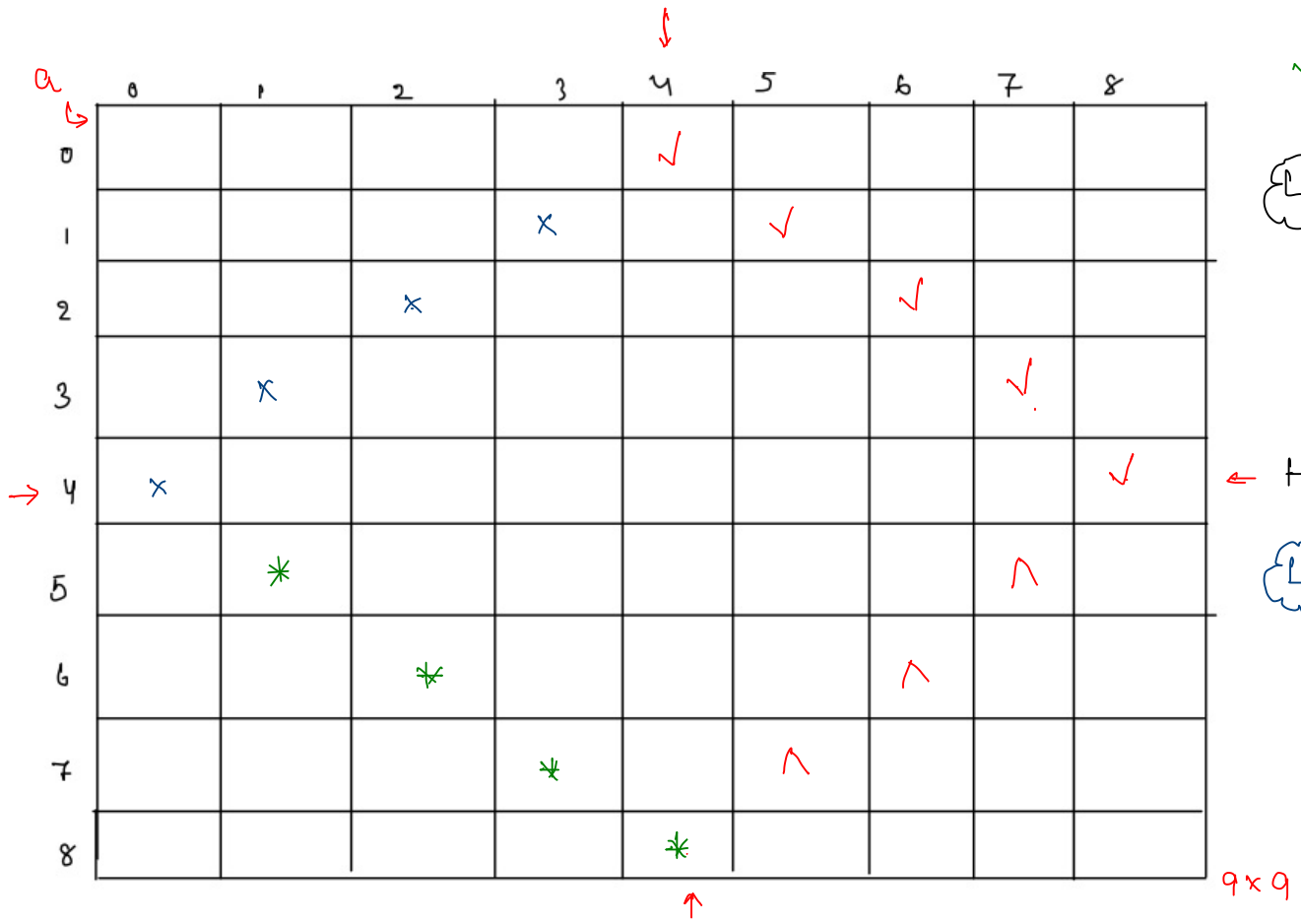
# 2. Traversal Techiques

## Diamond Traversal :-

### Description

Given a square matrix of odd length , print the matrix elements in the order shown in the figure :






$n \times n = 9 \times 9$ $\quad \lfloor \frac{9}{2} \rfloor = 4_{9/2}$

$\downarrow$ 3

L1

$a_{48}, a_{37}, a_{26}, a_{15}, a_{04}$

i ↵ ↳ j

i : 4 to 0 → $i = \frac{n}{2}$ to 0

j : 8 to 4 → $j = n-1$ to $\frac{n}{2}$

---

L2

$a_{13}, a_{22}, a_{31}, a_{40}$

i ↙ ↳ j

i : 1 to 4 → $i = 1$ to $\frac{n}{2}$

j : 3 to 0 → $j = \frac{n}{2} - 1$ to 0

---

L3

$a_{51}, a_{62}, a_{73}, a_{84}$

i ↙ ↳ j

i : 5 to 8 → $\frac{n}{2}+1$ to $n-1$

j : 1 to 4 → 1 to $n/2$

---

L4

$a_{75}, a_{66}, a_{57}$

i ↙ ↳ j

i : 7 to 5 → $n-2$ to $\frac{n}{2}+1$

j : 5 to 7 → $\frac{n}{2}+1$ to $n-2$

# Spiral Traversal :-

10-mins


7x7

$R_1$ → circular Trav.

$R_2$ → " "

$R_3$ → " "

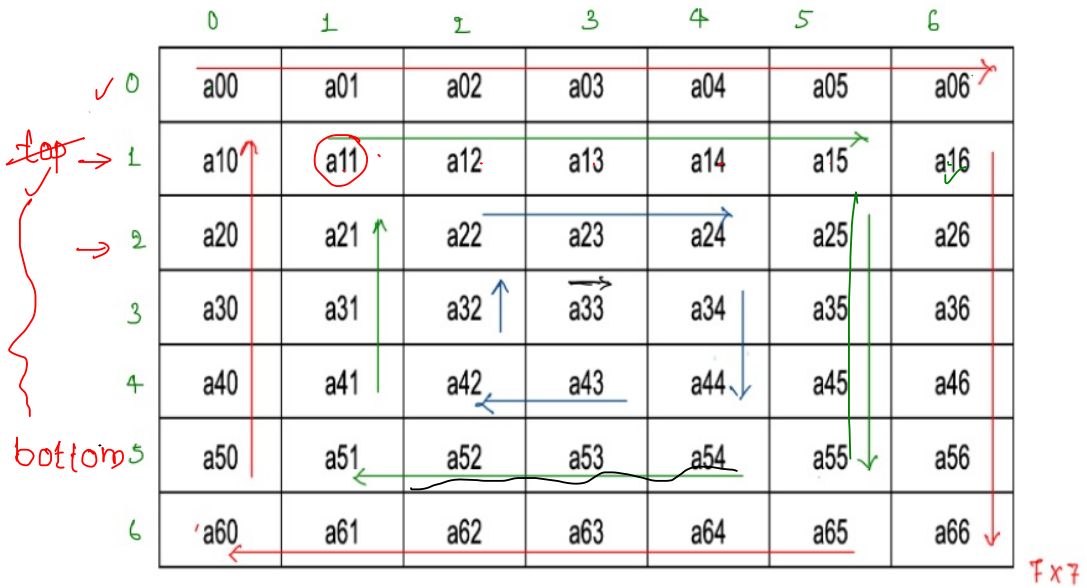
$R_4$ → " "

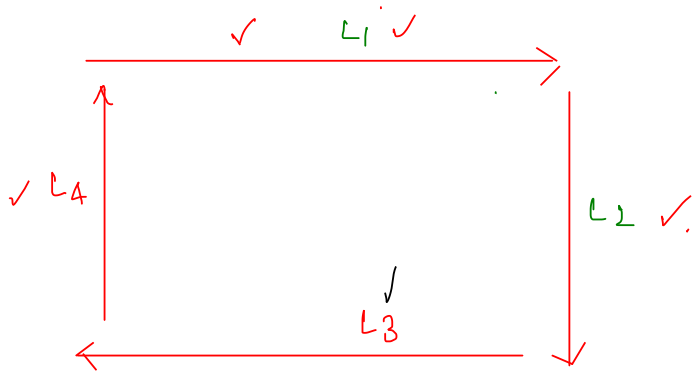
```
function spiralTraversal(arr,n)
{



}
```

WOW




nxn = 7x7

49

dry-run

```
function spiralTraversal(arr,n)
{
        top=0,bottom=n-1,left=0,right=n-1
        while(top<=bottom and left<=right)
        {
                for(j=left;j<=right;j++)
                {
                        print(arr[top][j])
                }
                top=top+1
                for(i=top;i<=bottom;i++)
                {
                        print(arr[i][right])
                }
                right=right-1
                for(j=right;j>=left;j--)
                {
                        print(arr[bottom][j])
                }
                bottom=bottom-1
                for(i=bottom;i>=top;i--)
                {
                        print(arr[i][left])
                }
                left=left+1
        }


}
```

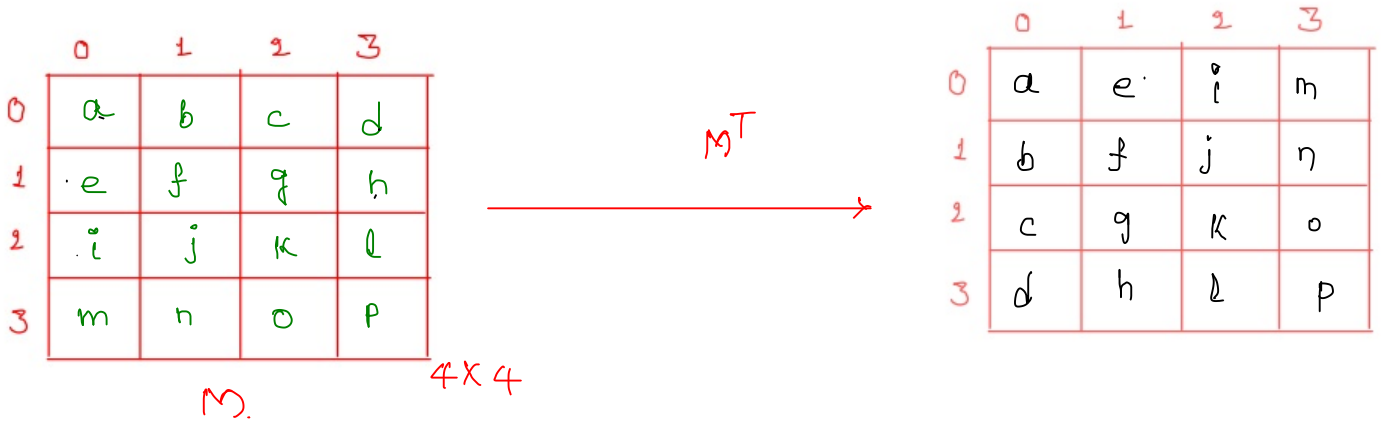
mxn

c = 0 to m*n

49

50 fail

Q) Find the Transpose of a Matrix [ N*N ]



original matrix

Transposed matrix

a

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | a | b | c | d |
| 1 | e | f | g | h |
| 2 | i | j | k | l |
| 3 | m | n | o | p |

4x4

o/p

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | a | e | i | m |
| 1 | b | f | j | n |
| 2 | c | g | k | o |
| 3 | d | h | l | p |

4x4

i=0

0 0 → 0 0

0 1 → 1 0

0 2 → 2 0

0 3 → 3 0

i=1

1 0 → 0 1

1 1 → 1 1

1 2 → 2 1

1 3 → 3 1

i=2

M[i][j] ⇄ M[j][i]

Swapping

```
for(i=0;i<n;i++)
{
        for(j=?;j<n;j++)
        {
                temp=arr[i][j]
                arr[i][j]=arr[j][i]
                arr[j][i]=temp
        }
}
```

swap a[i][j] and a[j][i]

i/p

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | a | b | c | d |
| 1 | e | f | g | h |
| 2 | i | j | k | l |
| 3 | m | n | o | p |

M

o/p

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | a | e | i | m |
| 1 | b | f | j | n |
| 2 | c | g | k | o |
| 3 | d | h | l | p |

b w.r.t e

i/p

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | a | b | i | m |
| 1 | e | f | j | h |
| 2 | c | g | k | l |
| 3 | d | n | o | p |

M

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | a | b | c | d |
| 1 | e | f | g | h |
| 2 | i | j | k | l |
| 3 | m | n | o | p |

M

Diff

a[i][j] swap a[j][i]

```
→ for(i=0;i<n;i++)
  {
        for(j=i;j<n;j++)
        {
              temp=arr[i][j]
              arr[i][j]=arr[j][i]
              arr[j][i]=temp
        }
  }
```

Correct Code for Transpose is as follows:

→ H/W

```
for(i=0;i<n;i++)
{
        for(j=i+1;j<n;j++)
        {
            1. temp=arr[i][j]
            2. arr[i][j]=arr[j][i]
            3. arr[j][i]=temp
        }
}
```

finding the Transpose of matrix.

# Q) Rotate N*N Matrix 90 degrees Clock Wise [ you should not use any extra array ]

i/p

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | a | b | c | d | e |
| 1 | f | g | h | i | j |
| 2 | k | l | m | n | o |
| 3 | p | q | r | s | t |
| 4 | u | v | w | x | y |

5X5

90°

| | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| 0 | u | p | k | f | a |
| 1 | v | q | l | g | b |
| 2 | w | r | m | h | c |
| 3 | x | s | n | i | d |
| 4 | y | t | o | j | e |

o/p

| | | | | |
|---|---|---|---|---|
| u | p | k | f | a |
| v | q | l | g | b |
| w | r | m | h | c |
| x | s | n | i | d |
| y | t | o | j | e |

i/p [M]

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | a | b | c | d | e |
| 1 | f | g | h | i | j |
| 2 | k | l | m | n | o |
| 3 | p | q | r | s | t |
| 4 | u | v | w | x | y |

after rotating 90° clock wise →

o/p

| | | | | |
|---|---|---|---|---|
| u | p | k | f | a |
| v | q | l | g | b |
| w | r | m | h | c |
| x | s | n | i | d |
| y | t | o | j | e |

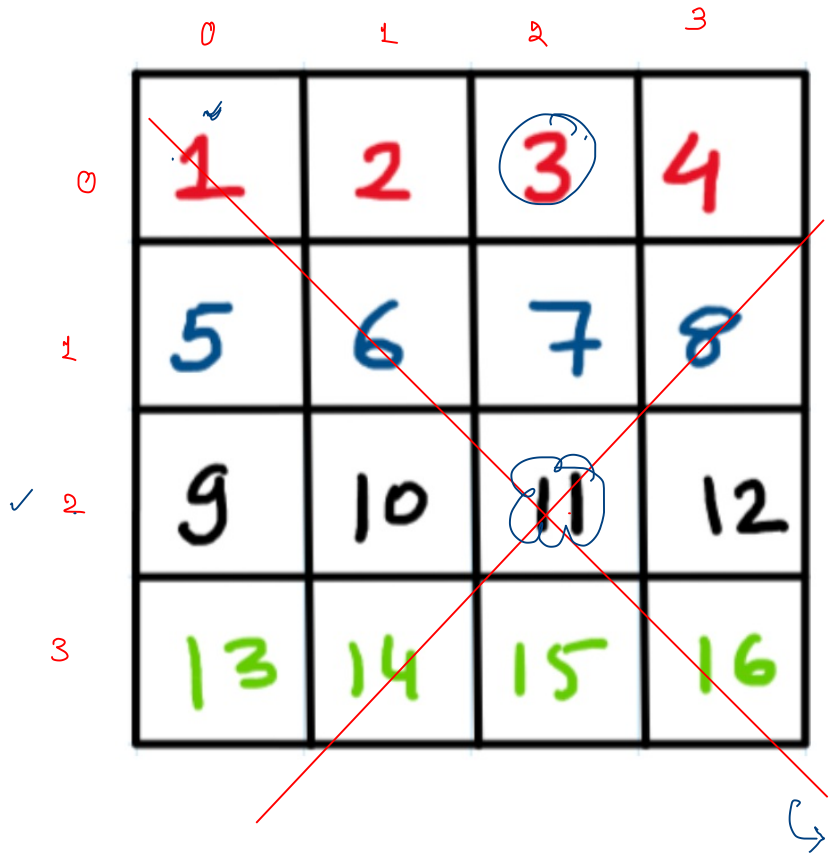
Step$_1$

/MT/

find transpose of a matrix

| | | | | |
|---|---|---|---|---|
| a | f | k | p | u |
| b | g | l | q | v |
| c | h | m | r | w |
| d | i | n | s | x |
| e | j | o | t | y |

Step$_2$ : reverse the rows




$x = 2$ $y = 2$

$k = 11$ →

0, 0

2, 2

starting : 1 1 → 0 0 ✓

one value is 0, Stop

+1

1 1 ✓

+1

2 2 ✓

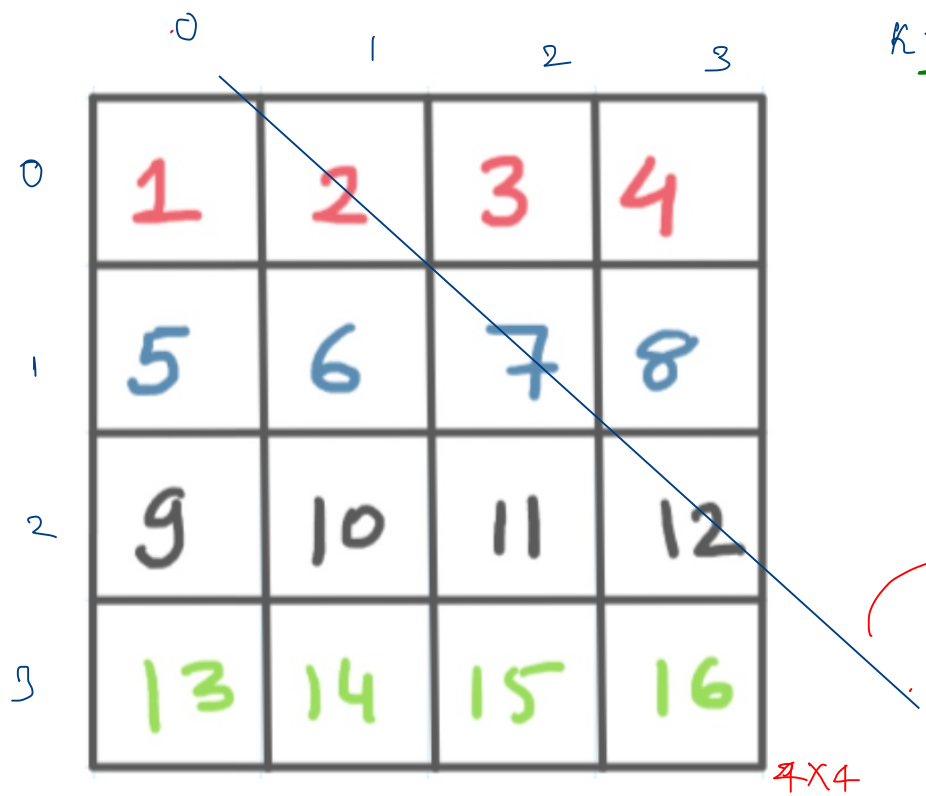
+1

3 3 ✓

+1

4 4 stop

ending :




K = 7

Starting : 0 1 → 1 2 → 2 3 → 3 4 Stop

finding (↓)

1 2

↓ -1

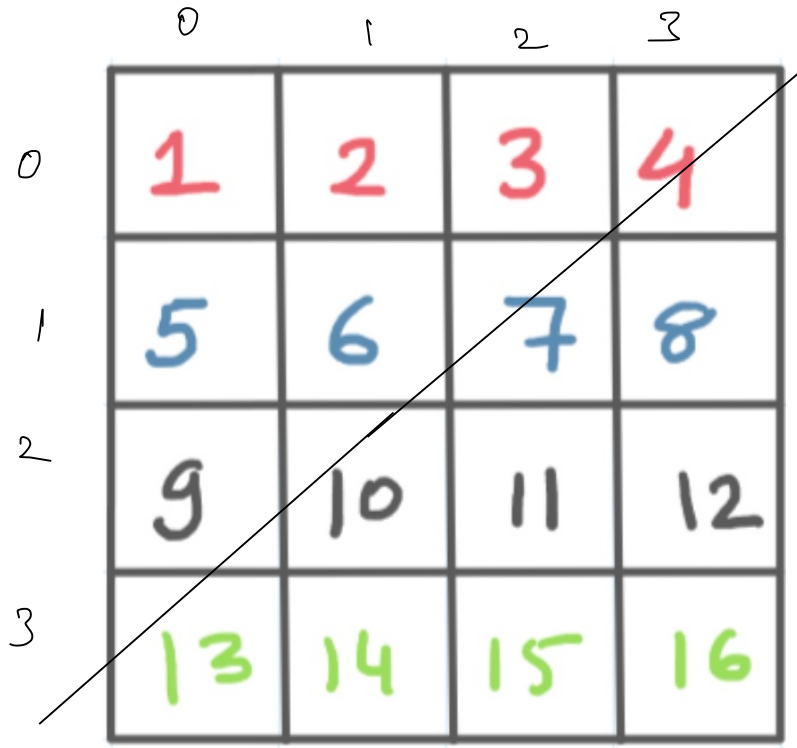
x: 0 1 y → Starting

↓ -1

-1 0 ← X

+1 +1 +1

0 0 → 3 3

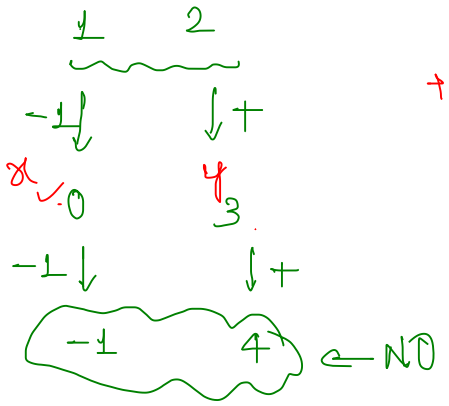


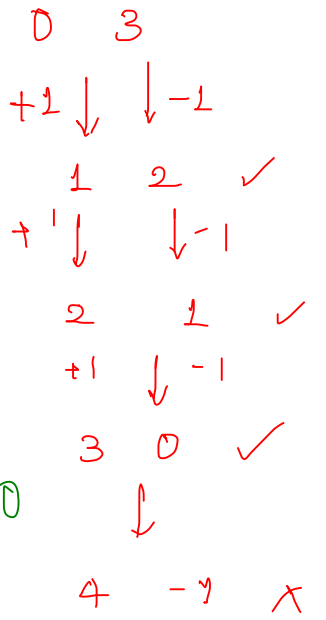

op

0 3, 1 2, 2 1, 3 0

K = 7

starting: ?




1 2 3
4 5 6
7 8 9
——Transpose——>
1 4 7
2 5 8
3 6 9
—-Reverse individual rows—->
7 4 1
8 5 2
9 6 3
(Resultant matrix)