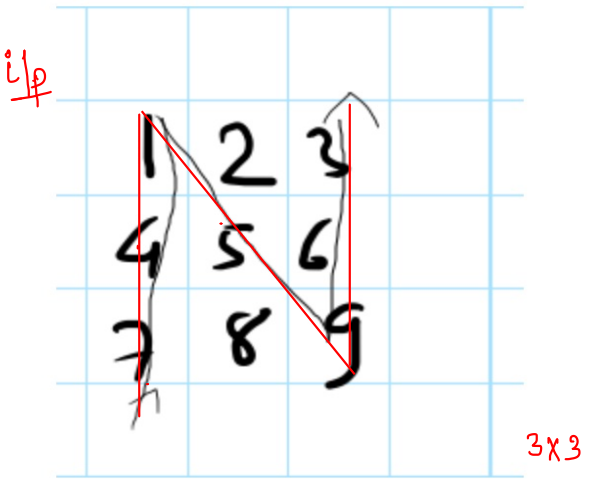
Sprint-1 [Day-2]

N traversal

Description

You are given a matrix of size n x n. Find the**N**traversal of the matrix. Refer the following figure for better understanding.



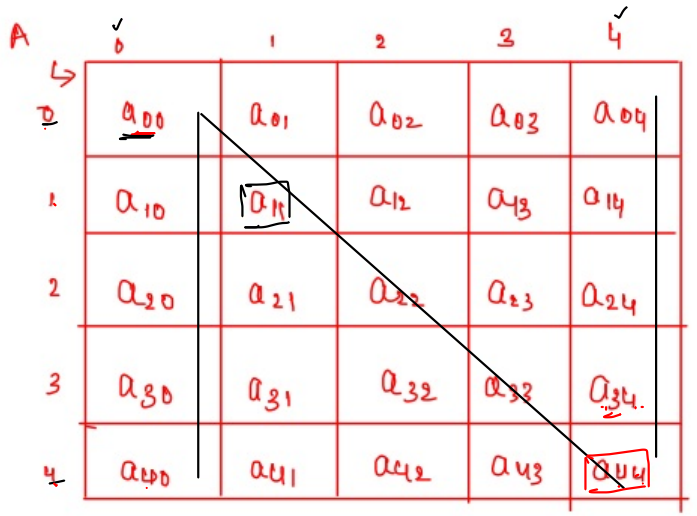
o/p

7, 4, 1, 5, 9, 6, 3

Input

The first line contains T, the number of test cases. The first line of each test case contains N, the size of the square matrix.

Next N lines contain N space separated integers, denoting the values of the matrix.



n = 5
0 to 4

5×5
n×n

op

i = 1 2 3 4

a11

a22

a33

a44

4

6 . . n-2 → 0

a34 3 → 2 → 1 → 0

a24

a14

a04

n = 5 . . n-2

-2

34

24

14

04

2D : [ ][ ]

```
function Ntraversal(arr,n)
{
        res=" "
    L1  for(i=n-1;i>=0;i--)
        {
                res=res+arr[i][0]+" "
        }

    L2  for(i=1;i<=n-1;i++)
        {
                res=res+arr[i][i]+" "
        }

    L3  for(i=n-2;i>=0;i--)
        {
                res=res+arr[i][n-1]+" "
        }

        return res; // console.log(res)
}
```

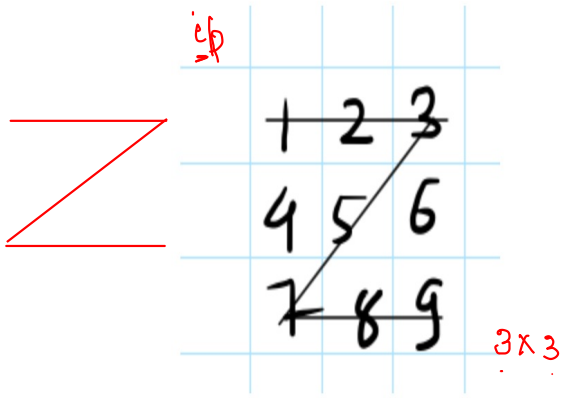
=> loops
↑ Unit-1

2 Z Traversal

-13:21:21

Description

Given a square matrix of size N x N. Print the Z traversal of the matrix. Refer the figure given below for better understanding.



op

1 2 3 5 7 8 9

Input

The first line of the input contains T, the number of test cases. The first line of each test case contains N, the dimension of the square matrix.

Next N lines contains N space separated integers, denoting the values of the matrix.

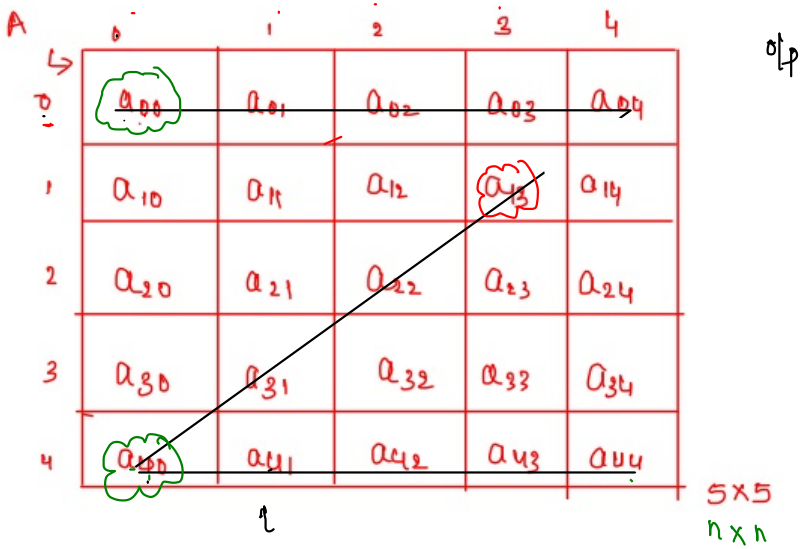
Constraints

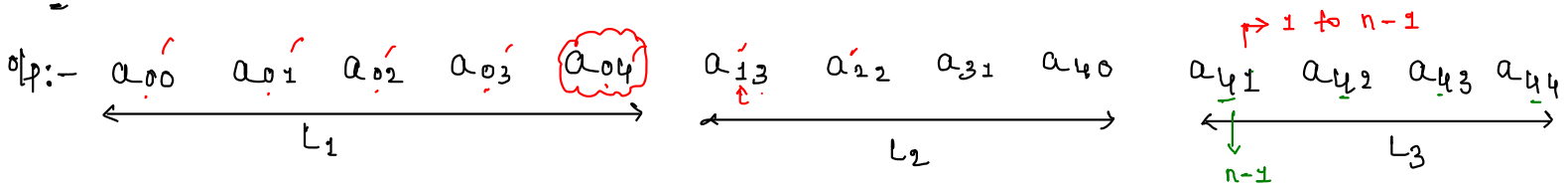
1 <= T <= 10

1 <= N <= 500

1 <= A[i][j] <= 1000

Output





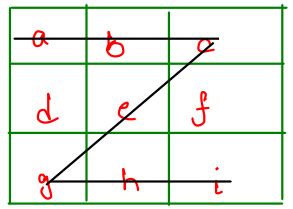
L_2

row	col
1	3 ← n-2
2	2
3	1
4	0 ← 0

(1 to n-1)

```
function Ztraversal(arr,n)
{
        res=" "
L1      for(i=0;i<=n-1;i++)
        {
                res=res+arr[0][i]+" "
        }
        i=1,j=n-2
L2      while(i<=n-1 && j>=0)
        {
                res=res+arr[i][j]+" "
                i++
                j--
        }
L3      for(i=1;i<=n-1;i++)
        {
                res=res+arr[n-1][i]+" "
        }
        return res
}
```

```
for(i=1,j=n-2; i<=n-1 && j>=0; i++,j--)
{
        res=res+arr[j][j]+" "
}
```



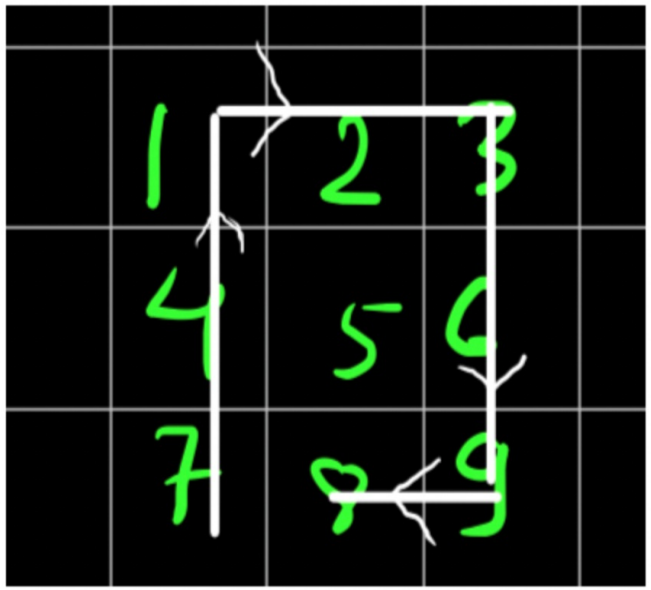
Circular Traversal

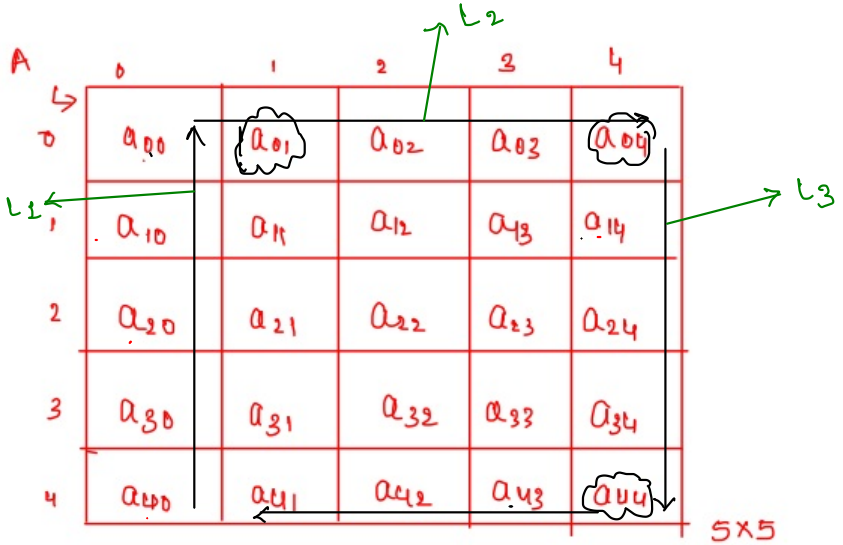
-13:6:59 Edit

Description

Given a square matrix, you have to find the reverse U traversal of the matrix. Refer the sample I/O for better understanding. Refer the given figure for better understanding.

Note: No element should be visited more than once.





L_1 : a_{40} a_{30} a_{20} a_{10} a_{00} ⇒ col : 0

row : n-1 → 0

```
res=" "
for(i=n-1;i>=0;i--)
{ left
        res=res+arr[i][0]+" "
}
```

Nested Loop

L_2 : a_{01} a_{02} a_{03} a_{04} ⇒ row : 0

col : 1 → n-1

```
for(i=1;i<=n-1;i++)
{
        res=res+arr[0][i]+" "
}
```

L_3 : a_{14} a_{24} a_{34} a_{44} ⇒ col : n-1

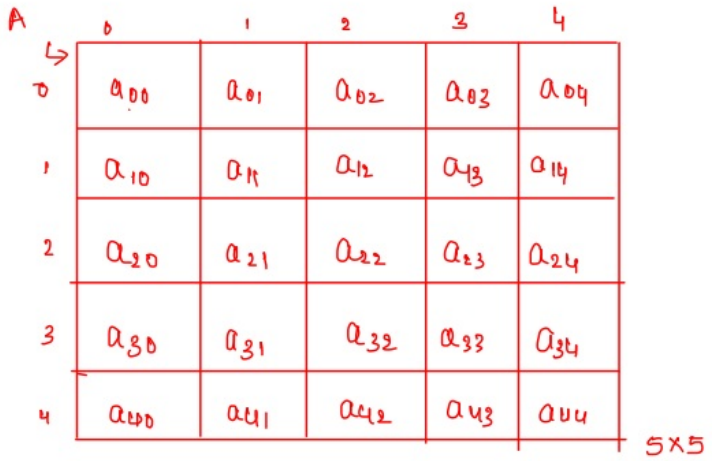
row : 1 → n-1

```
for(i=1;i<=n-1;i++)
{
        res=res+arr[i][n-1]+" "
}
```

L_4 :- a_{43} a_{42} a_{41} ~~a_{40}~~ row : n-1

col : n-2 → 1

```
for(i=n-2;i>=1;i--)
{
        res=res+arr[n-1][i]+" "
}
```



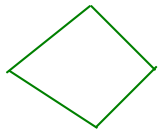

```
temp=" "
for(i=n-1;i>=0;i--)
{
        temp=temp+A[i][0]+" "
}
```

```
for(j=1;j<=n-1;j++)
{
        temp=temp+A[0][j]+" "
}
```

```
for(i=1;i<=n-1;i++)
{
        temp=temp+A[i][n-1]+" "
}
```

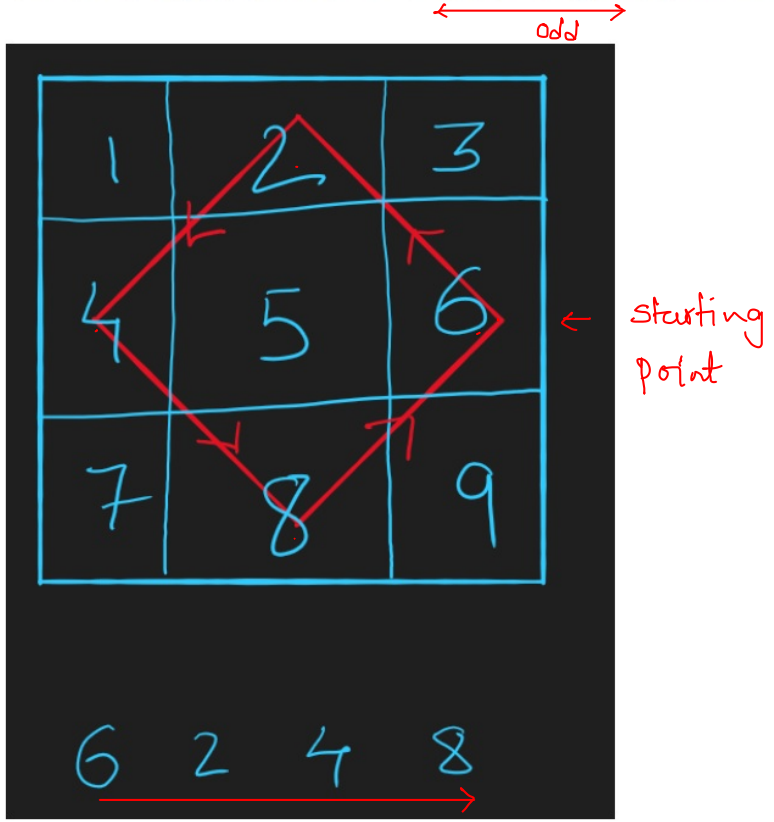
```
for(j=n-2;j>=1;j--)
{
        temp=temp+A[n-1][j]+" "
}
```

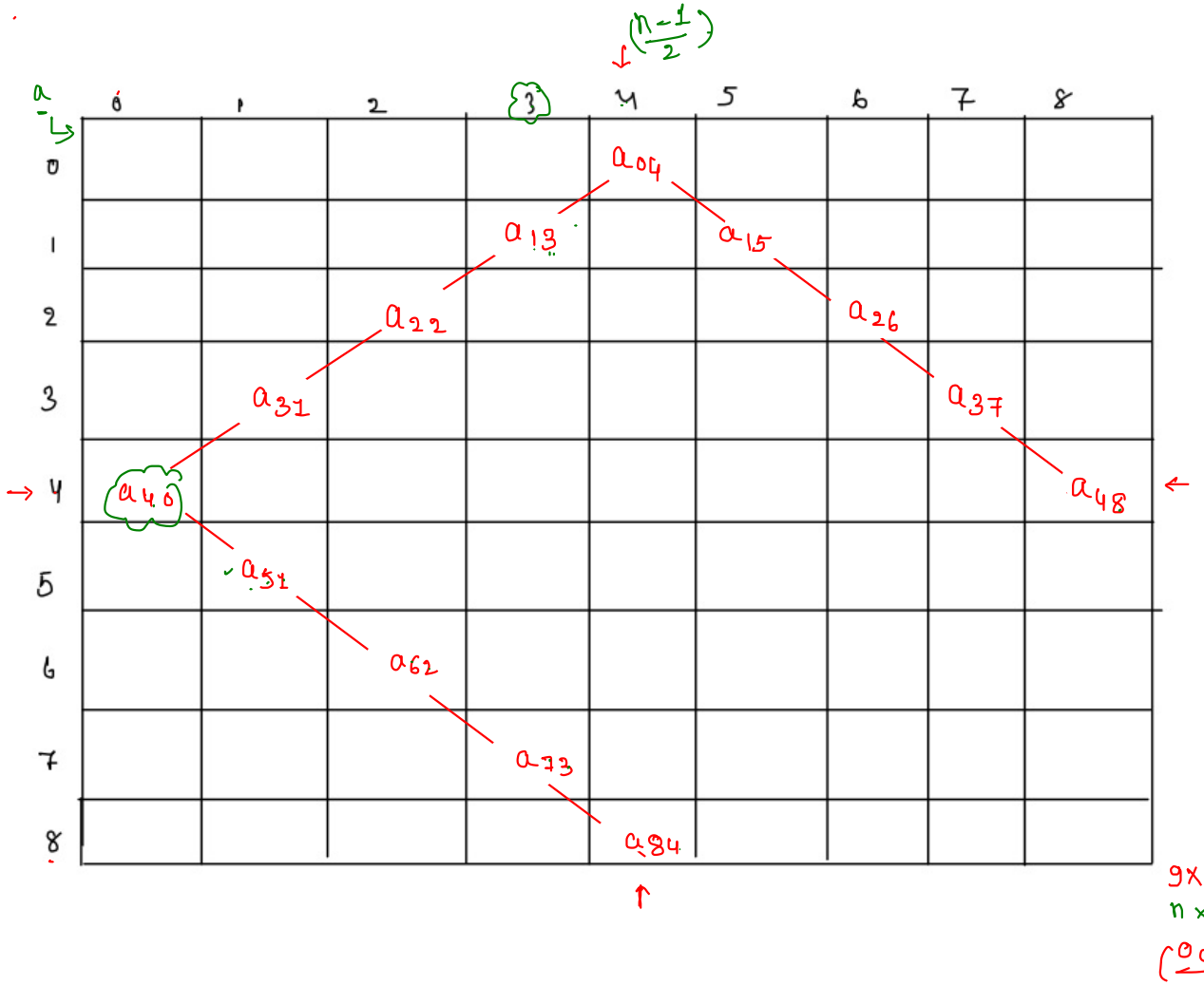
Diamond Traversal



Description

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





$n=9 \Rightarrow 9 \times 9$

floor : $\lfloor \; \rfloor$

$\frac{9}{2} = 4.5 \Rightarrow \lfloor 4.5 \rfloor = 4$

L_1 :- a[i][j]

i : $4 \to 0$, $\left(\frac{n-1}{2}\right) \to 0$

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

L_2 :- a[i][j]

i : $1 \to 4$, $1 \to \frac{n-1}{2}$

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

L_3 :- a[i][j]

i : $5 \to 8$, $\frac{n-1}{2}+1 \to n-1$

j : $1 \to 4$, $1 \to \frac{n-1}{2}$

L_4 ✓