

S	M	T	W	T	F	S
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

## Matrix

SEPTEMBER

'24

37th Week  
257-109

13

FRIDAY

→ A matrix represents a collection of numbers arranged in order of rows & columns. It is necessary to enclose the elements of matrix in parenthesis.

1	2	3
4	5	6
7	8	9

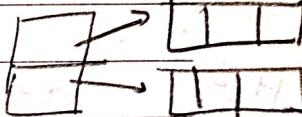
row

column

box

Order of matrix = No. of rows x No. of columns

→ A 2D array is simply a list of lists in Python:



arr = [[1, 2, 3], [4, 5, 6]]

for i in arr:

    for j in i:

        print(j, end=" ")

    4 5 6 7 8

    print()

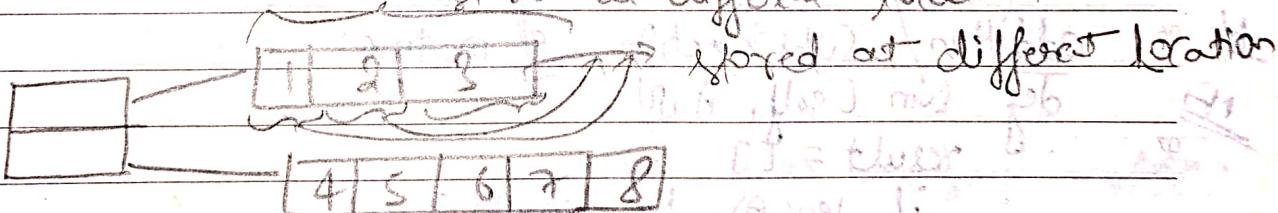
print('No. of rows', len(arr))

print('Count in 1st row!', len(arr[0]))

print('Count in 2nd row!', len(arr[1]))

References of these lists stored in contiguous location, but the list items themselves might be stored in different location.

Stored at different location



alternate way  
to traversal

arr = [[1, 2, 3], [4, 5, 6, 7, 8]]

for i in range(len(arr)):

    for j in range(len(arr[i])):

        print(arr[i][j], end=" ")

    1 2 3

    4 5 6 7 8

    print()

He travels fastest who travels alone

SEPTEMBER

14

'24

37th Week  
258-108

S	M	T	W	T	F
1	2	3	4	5	6
8	9	10	11	12	13
15	16	17	18	19	20
22	23	24	25	26	27
29	30				

SATURDAY

# User Specified Dimensions ~~Not Recommended~~

M-1 08

cols = 4

09

arr = [[0]\*cols]\*rows

$$\begin{cases} [1, 0, 0, 0] \\ [1, 0, 0, 0] \\ [1, 0, 0, 0] \end{cases}$$

10

for i in arr:  
    print(i)

11

arr = [[0 for j in range(cols)] for i in range(rows)]

12

arr[0][0] = 1  
for i in arr:  
    print(i)
$$\begin{cases} [1, 0, 0, 0] \\ [0, 0, 0, 0] \\ [0, 0, 0, 0] \end{cases}$$

# 02 Passing 2D array as a argument

03

def print\_mat(mat):  
    m = len(mat) # No. of rows  
    n = len(mat[0]) # No. of columns  
    for i in range(m):

04

05

# 06 addition / subtraction of matrix

M-1

def sum(self, A, B)

07

result = []

if len(A) != len(B):

return result

for i in range(len(A)):

res = []

if len(A[i]) != len(B[i]):

return result

for j in range(len(B[i])):

res.append(A[i][j] + B[i][j])

result.append(res)

return result

However long the night, the dawn will break

	S	M	T	W	T	F	S
OCT 2024		1	2	3	4	5	
6	7	8	9	10	11	12	
13	14	15	16	17	18	19	
20	21	22	23	24	25	26	
27	28	29	30	31			

SEPTEMBER

'24

38th Week  
259-107

15

SUNDAY

## # Print matrix in Snake pattern.

```

7/p= mat = [{ { 10,20,30,40 }, 10→20→30→40
08 { 15,25,35,45 }, 15←25←35←45
09 { 27,29,37,48 }, 27→29→37→48
{ 32,33,39,50 } }; 32←33←39←50

```

10 def printsnake(mat):

m = len(mat)

n = len(mat[0])

for i in range(m):

if i%2 == 0:

for j in range(n):

print(mat[i][j], end = " ")

else:

for j in range(n-1, -1, -1):

print(mat[i][j], end = " ")

## # Transpose of matrix

def transpose(A,B):

for i in range(n):

for j in range(N):

B[i][j] = A[j][i]

Just changing upper triangular matrix to lower triangular matrix.

def transpose(A,B):

for i in range(N):

for j in range(i+1, N):

A[i][j], A[j][i] = A[j][i], A[i][j]

TC = O(n<sup>2</sup>)

SC = O(1)

SEPTEMBER

16

'24

38th Week  
260-106

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

MONDAY

## # Spiral Traversal of matrix!

08

Given a 2d array, print it in spiral form,

09  $\text{if } \beta = [[1, 2, 3, 4],$  $[5, 6, 7, 8],$ 10  $[9, 10, 11, 12],$  $[13, 14, 15, 16]]$ 

11

 $\text{so } \beta = [1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5, 6] \Rightarrow \text{Top} \rightarrow \text{Right}$ 

12

def spiral (mat):

01 R = len(mat)

02 C = len(mat[0])

03 top = 0

04 left = 0

05 bottom = R-1

06 right = C-1

07 while (top &lt;= bottom) and (left &lt;= right):

08     for i in range (left, right+1):

09         print (mat [top] [i]), end = " "

10         top += 1

11         for i in range (top, bottom-1):

12             print (mat [i] [right]), end = " "

13         right -= 1

14         bottom -= 1

15         for i in range (right, left-1, -1):

16             print (mat [bottom] [i]), end = " "

17         bottom -= 1

18         if (left &lt;= right):

19             for i in range (bottom, top-1, -1):

20                 print (mat [i] [left]), end = " "

He who lives by the sword shall die by the sword

left += 1.

S	M	T	W	T	F	S
OCT 2024	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

'24

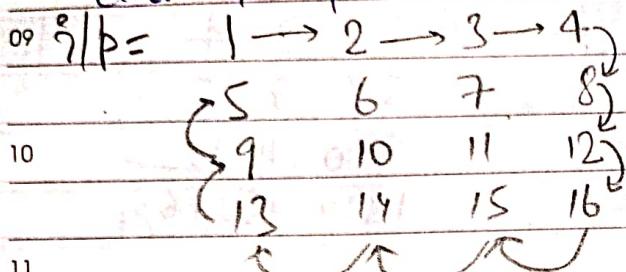
17

38th Week  
261-105

TUESDAY

## # Matrix Boundary Traversal:-

08 The task is to print the boundary element of the given matrix in clockwise form.



09  $\text{g/p} = 1, 2, 3, 4, 8, 12, 6, 15, 14, 13, 9, 5$

## # Naive approach:-

- 12 The simplest approach to solve this problem is to traverse the given matrix and check if the current element is the boundary element or not. If found to be true, then print the element.
- 01 Point the first row of matrix.
- 02 Point the last column of matrix except last row.
- 03 Point the last row of matrix except last column.
- 04 Point the first column of matrix except the 1st & last row.

03 def Traversal(mat):  
    R = len(mat)  
    C = len(mat[0])

04     if R == 1:

05         for i in range(C):

06             print(mat[0][i], end=" ")

07     elif C == 1:

08         for i in range(R):

09             print(mat[i][0], end=" ")

10     else:

11         for i in range(C):

12             print(mat[0][i], end=" ")

13         for i in range(1, R-1):

14             print(mat[i][C-1], end=" ")

15             for i in range(C-2, -1, -1):

16             print(mat[R-1], [i], end=" ")

17         for i in range(R-2, 0, -1):

18             print(mat[i][0], end=" ")

He who pays the piper calls the tune

19             print(mat[0][0], end=" ")

# SEPTEMBER

18

38th Week  
262-104

S	M	T	W	T	F	S	
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30						SEP 2024

## WEDNESDAY

## # Rotate a matrix! -

The task is to rotate the matrix in anti-clock wise direction by  $90^\circ$  degrees.

$$\begin{array}{r} \frac{9}{\cancel{1}} \\ \times 4 \\ \hline 369 \end{array}$$

$\frac{1}{2}$	$\frac{3}{4}$	$\frac{5}{6}$	$\frac{7}{8}$
$\frac{9}{10}$	$\frac{11}{12}$		
$\frac{13}{14}$	$\frac{15}{16}$		

$\rightarrow$  first row of destination  $\rightarrow$  Last column of source.  $\Rightarrow$  48 12 16  
 $\rightarrow$  second row of destination  $\rightarrow$  second last column of source.  $\Rightarrow$  3 7 11 15  
 $\rightarrow$  last row of destination  $\rightarrow$  first column of source.  $\Rightarrow$  5 9 13

~~Naive Approach~~ The idea is to create a temporary matrix of same dimension as that of the original matrix and copy the original matrix into this temporary matrix. Finally, replace entries in the original matrix one by one by column of the temporary matrix from last to first.

def rotate(mat):  
    n = len(mat)

for about 1/2 hour. Here is dry run

`temp = [0]*n for i in range(n)]: if for i == 0: temp[0] = mat`

for i in range(n):  
    for j in range(n):

$$\text{for } j \text{ in range } (n): \quad (4-i-j)j = 1: \quad (210) \quad (0,1)$$

$$07 \quad \text{temp}[n-j-1][i] = \text{mat}[i][j] \quad (n-2-1) \quad j=2 \quad (1,0) \quad (0,1)$$

$$\text{temp} [n-1 \ J-1] [i] = \text{mat} [i][j] [n-2-i] \ j=2:1 (1,0) \rightarrow (0,1)$$

for i in range(n):  
 b[i] = f((i-1)\*J+1)

for j in range(n): print((j+1), end="")

$\text{mat}[i][j] = \text{temp}[i][j]$  } Last column become first row

for i in range(n): second last column becomes

for j in range(n):  
 print(f'{a[j]} {b[j]} {c[j]}')

```
print(mat[i][j], end = " ")
```

Honey catches more flies than vinegar.

Honey catches more flies than vinegar

	S	M	T	W	T	F	S
OCT 2024							
	6	7	8	9	10	11	12
	13	14	15	16	17	18	19
	20	21	22	23	24	25	26
	27	28	29	30	31		

## SEPTEMBER

'24

38th Week  
263-103

19

## THURSDAY

## Efficient Solution

- 08 ⑩ Find transpose of matrix and O(n<sup>2</sup>)  
 ⑪ Reverse individual Columns and O(1).  
 SC

09	1 2 3 4	1 5 9 13	Reverse Column	4 8 12 16
	5 6 7 8	2 6 10 14		3 7 11 15
10	9 10 11 12	3 7 11 15		2 6 10 14
	13 14 15 16	4 8 12 16		1 5 9 13

06 # Search in Row wise & column wise Sorted matrix

- Given a  $n \times n$  matrix and an integer  $m$ , find the position of  $m$  in the matrix if it is present. Otherwise, print "Element not found". Every row of the matrix is sorted in increasing order. The

if:  $\text{mat}[\mathcal{Y}][4] = \{10, 20, 30, 40\}$ ,

$$\begin{array}{l} \{15, 25, 35, 45\}, \\ \{27, 29, 37, 48\}, \\ \{32, 33, 34, 35\} \end{array}$$

$\circ/\rho$ : found at  $(\frac{5}{2}, 1)$

**He who wills the end wills the means**

$$\text{if } \beta : \text{med}[\{4\}][4] = \{\{10, 20, 30, 40\}\},$$

200

{ 15, 25, 35, 75 },  
{ 27, 29, 37, 48 } ,  
{ 31, 33, 34, 35 }

{ 33, 33, 34, 35 }

$\beta$  = not found

SEPTEMBER

20

'24

38th Week  
264-102

S	M	T	W	F	S
1	2	3	4	5	6
8	9	10	11	12	13
15	16	17	18	19	20
22	23	24	25	26	27
29	30				

FRIDAY  
Naive Approach

The simple idea is to traverse the array &amp; search element one by one.

08

def search(mat, n):

09

if (n == 0):

return -1

TC = O(n^2)

10

for i in range(n):

SC = O(1)

11

for j in range(n):

if mat[i][j] == n:

print("Element found at (", i, ", ", j, ")")

12

return 1

return 0

Efficient Solution

→ Begin from the top right corner

02 def search(mat, m):

→ If n is large, Point position &amp; return

r = len(mat)

→ If n is smaller, move left.

03 c = len(mat[0])

→ If n is greater, move down.

i = 0

04 j = c - 1

while i &lt; r and j &gt;= 0:

05 if mat[i][j] == n:

Print("found at (", i, ", ", j, ")")

06 return 1

else if mat[i][j] &gt; n:

07 j -= 1

Point("Not found")