



Today's agenda

- ↳ Intro to 2D Arrays

- ↳ Print matrix row wise

- ↳ Print matrix Colwise

- ↳ Print matrix in wave form

- ↳ Arraylist



AlgoPrep



//Intro

Sub0 → math
Sub1 → english
Sub2 → science

Column

	0	1	2
0	99	90	96
1	90	95	95
2	75	80	30
3	100	100	99
4	100	100	101

→ 2D Array

Syntax:

↳ `int[][] arr = new int[5][3]`

0 1 2

0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

5*3

`s.o.p(arr[4][1]);`

`arr[4][2] = 100;`

↳ $5 \times 3 = 15$ elements

No. of rows: `arr.length`

No. of columns: `arr[0].length`



Q) Print matrix rowwise

↳ Print the given $\text{mat}[n][m]$ rowwise.

$\text{arr}[4][5]?$

	0	1	2	3	4
0	10	20	30	40	50
1	60	70	80	90	100
2	110	120	130	140	150
3	160	170	180	190	200

4x5

10 20 30 40 50
60 70 80 90 100
110 120 130 140 150
160 170 180 190 200

$i \quad j$

0,0	1,0	2,0	3,0
0,1	1,1	2,1	3,1
0,2	1,2	2,2	3,2
0,3	1,3	2,3	3,3
0,4	1,4	2,4	3,4

$i =$ 0 1 2 3

$j \rightarrow$ {0,1,2,3,4} {0,1,2,3,4} {0,1,2,3,4}



Java Code

```
public static void PrintRowwise (int[] [] arr) {  
    int n = arr.length; // row count  
    int m = arr[0].length; // col count
```

T.C: $O(n \times m)$
S.C: $O(1)$

```
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < m; j++) {  
            System.out.print (arr[i][j] + " ");  
        }  
        System.out.println();  
    }  
}
```



// Input

Run Code

Untitled

Save

Java

Output: Finished

Clear Console

```
1 // "static void main" must be defined in a public class.
2 public class Main {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5
6         int n = scn.nextInt();//row no
7         int m = scn.nextInt();//col no
8         int[][] arr = new int[n][m];
9
10        for(int i=0;i<n;i++){
11            for(int j=0;j<m;j++){
12                // System.out.print(arr[i][j]);
13                arr[i][j] = scn.nextInt();
14            }
15        }
16
17        for(int i=0;i<n;i++){
18            for(int j=0;j<m;j++){
19                System.out.print(arr[i][j]+" ");
20            }
21            System.out.println();
22        }
23
24        int[][] arr1 ={{10,20,30,40},
25                      {50,60,70,80},
26                      {100,200,300,400}};
27
28        System.out.println(arr1.length);
29        System.out.println(arr1[0].length);
30    }
31 }
```

Finished in 179 ms

10 20 30 40
50 60 70 80
90 100 120 130
3
4

stdin

3 4
10 20 30 40
50 60 70 80
90 100 120 130

Share Live

Add Snippet



Algorithm



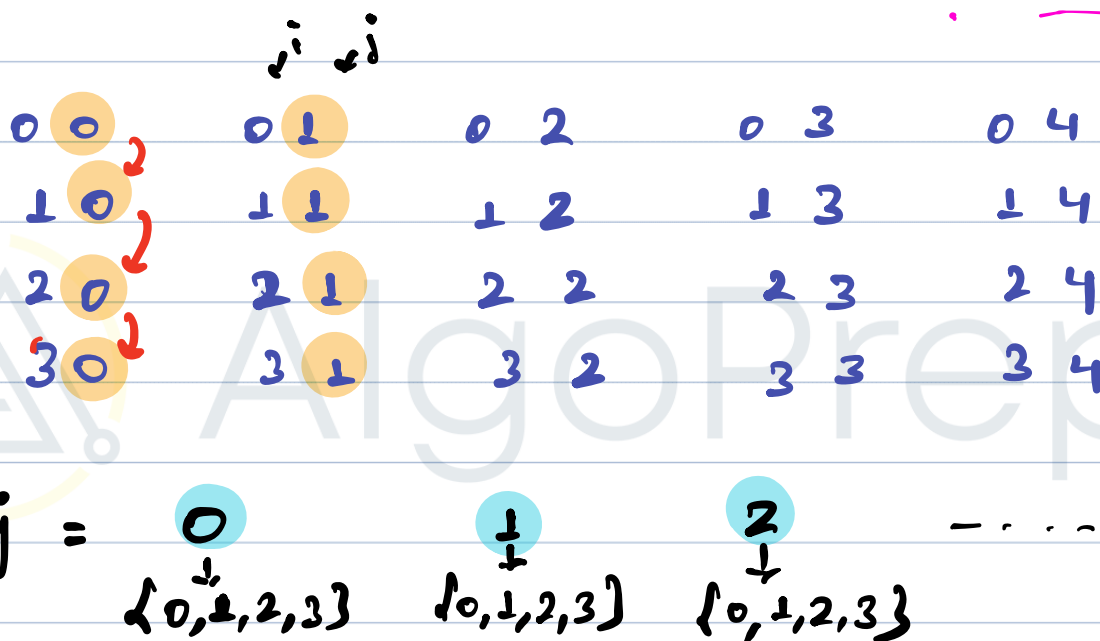
Q) Print matrix colwise

↳ Print the given $mat[n][m]$ colwise.

$arr[4][5]$:

	0	1	2	3	4
0	10	20	30	40	50
1	60	70	80	90	100
2	110	120	130	140	150
3	160	170	180	190	200

10 60 110 160
20 70 120 170
30 80 130 180
...



Pseudo code

```
Public Static void columnwise (int arr[n][m]){
```

```
    for (int j=0; j<m; j++){
```

```
        for (int i=0; i<n; i++){
```

```
            System.out.print (arr[i][j] + " ");
```

```
        }
```

```
        System.out.println();
```

```
    }
```

```
}
```

T.C: $O(n \times m)$

S.C: $O(1)$

Break till 10:25 PM



Q) Print matrix in wave form

↳ Print the given $mat[n][m]$ in wave form.

ans[4][5]?

0	10	20	30	40	50	L-R → 10 20 30 40 50
1	60	70	80	90	100	R-L → 100 90 80 70 60
2	110	120	130	140	150	L-R → 110 120 130 140 150
3	160	170	180	190	200	R-L → 200 190 180 170 160

L-R → {0, 2, 4, 6, ...} → even rows

R-L → {1, 3, 5, 7, ...} → odd rows

Pseudo Code

```
for (int i=0; i<n; i++) {  
    if (i%2==0) {  
        for (int j=0; j<m; j++) {  
            System.out.print (ans[i][j]+" ");  
        }  
    }  
    else {  
        for (int j=m-1; j>=0; j--) {  
            System.out.print (ans[i][j]+" ")  
        }  
    }  
    System.out.println();  
}
```

int \rightarrow Integer
long \rightarrow Long



ArrayList \approx dynamic array
 \hookrightarrow you don't need to provide the size of this array.

Syntax

\rightarrow `ArrayList < Integer > al = new ArrayList < > ();`

0	1	2
10	20	30

`al.size();` $\rightarrow 0$
 \downarrow
No. of elements in ArrayList

add an element at the end

`al.add(10);` \rightarrow T.C: $O(1)$

`al.add(20);`

`al.add(30);`

// get an element

\hookrightarrow `al.get(idn);`

T.C: $O(1)$

// remove an index

`al.remove(idn);`

remove from mid types
 $\hookrightarrow O(N)$

remove from last idn
 $\hookrightarrow O(1)$

al:

0	1	2	3	4
10	20	30	40	50

(Note: In the original image, the value 30 at index 2 is crossed out with a red line, and arrows indicate a shift of elements to the right.)

\hookrightarrow `al.remove(2)`

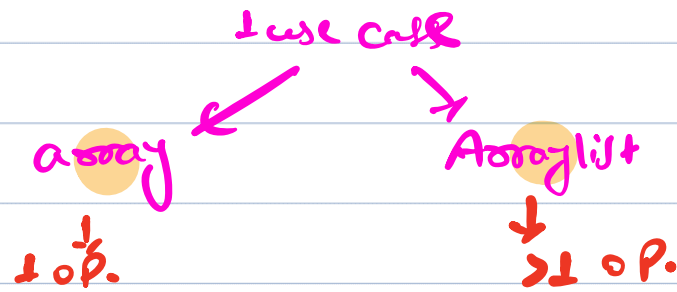
// change the index val

\hookrightarrow `al.set(idn, value);` \rightarrow T.C: $O(1)$

\hookrightarrow ArrayList in background is just array being smartly used.

$O(1)$
↑ ↑ 1
int n = 10;

$O(1)$ ← 5 operations
al.set(idn, val);



AlgoPrep