2D Arrays

NOTE: select Java 8 (array support)

| | |
|---|---|
| Nikhil D | Pranjul Kesharwani |
| Manohar A N | Deepika |
| Bhushan Ganesh Shelar | Shambhavi Belligatti |
| Vishnu Kumar P | Manisha |
| Hemanth T | Dhanashree Sagane |
| Naveen | Shivam Shiv |
| Yoga Nand Sattvik | Dhruv |
| Srikant Kumar Pratihary | Amreshwar |
| Hemant Verma | Anil |
| Mohona Paul | Pushpa |
| Chandra Shekhar Bhatt | M S Haseeb Khan |
| Ved Verma | Deepshikha Arora |
| Bhavesh Pandey | Iti |
| Nutan | Shaik Lal Jan Basha |
| Dev Raj Gaur | jeevanantham |
| Saurabh | Sai Kiran JNR |
| Harshad Marathe | Mahesh |
| Chirag | Shivanand Patil |
| Dhasthagiri Reddy | Rishav Ghosh |
| Farzana kauser | Rahul |

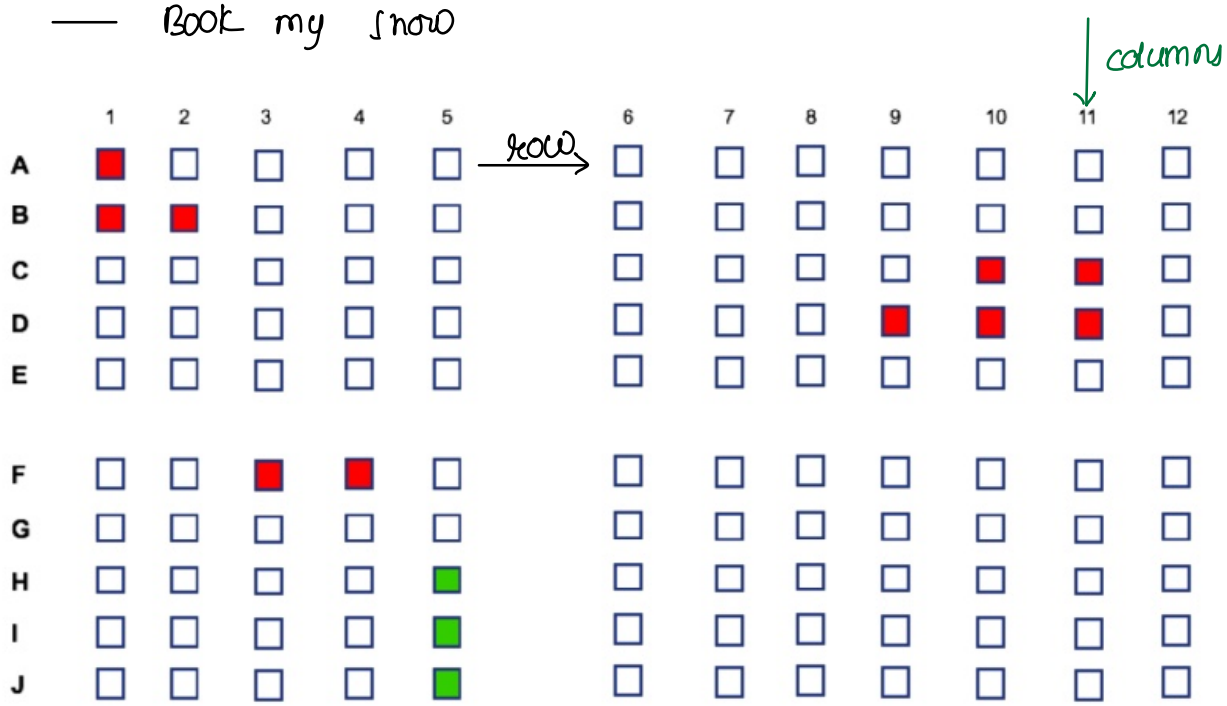
2D Array ⟶ Array of Arrays

Properties of 2D Array {matrix}

- Contains same type of data
- Size is known length & Breadth
- Each array is of same length

Real time application of 2D array.

- chess
- Snake and ladder game
- Book my show

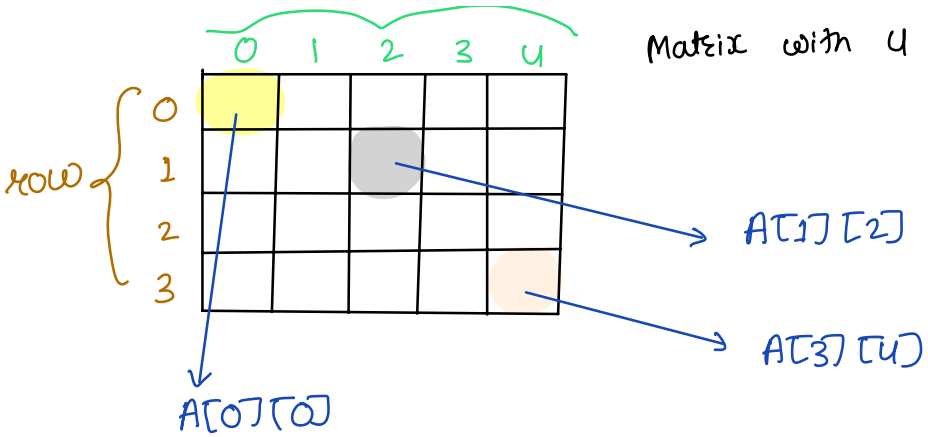



How to initialize 2D Array or matrix ?

⟶ `int[][] A = new int [4] [5] ;`

no. of rows  no. of columns

columns

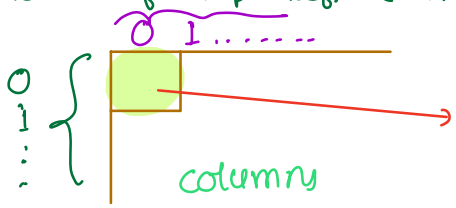



Matrix with 4 rows & 5 columns

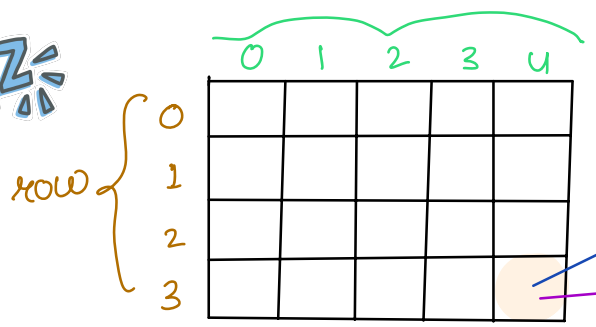
**QUIZ:** How to create a matrix with 5 columns & 7 rows ?

```
int[][] A = new int[7][5];
```

**QUIZ:** If you have matrix of size M * N. Index of top left corner ?




**QUIZ:**




M ⟶ Rows 4

N ⟶ Cols 5

Q> Print top row of matrix

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 6 | 7 | 8 | 9 | 10 |
| 2 | 11 | 12 | 13 | 14 | 15 |
| 3 | 16 | 17 | 18 | 19 | 20 |

→ 1 2 3 4 5

NOTE: for the top row index of row will always be 0

```
void printTopRow(int[][] A) {
    int R = A.length;      // 4
    int C = A[0].length;   // 5

    for(int c = 0; c < C; c++) {
        S.O.P(A[0][c] + " ");
    }
}
```

0th row or top row

Q> Print left most col of matrix

C

| R \ C | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 6 | 7 | 8 | 9 | 10 |
| 2 | 11 | 12 | 13 | 14 | 15 |
| 3 | 16 | 17 | 18 | 19 | 20 |

→ 1 6 11 16

Col index is always 0

```
void printLeftCol(int[][] A) {
    int R = A.length;      // 4
    int C = A[0].length;   // 5

    for(int r=0; r<R; r++) {
        S.O.p(A[r][0] + " ");
    }
}
```

0th col

row keeps changing

---

Visualise 2D array

⟶ Visualise 1D array.

```
int[] A = new int[4]
```

A = { 1 , 2 , 3 , 4 }
0 1 2 3

```
int[][] A = new int[3][2]
```

NOTE: all $A_1, A_2, A_3$ should be of same length

A =

| | 0 | 1 |
|---|---|---|
| 0 | 1 | 2 |
| 1 | 3 | 4 |
| 2 | 5 | 6 |

$A = \{ A_1, A_2, A_3 \}$

$A_1 = \{1, 2\}$ $A_2 = \{3, 4\}$ $A_3 = \{5, 6\}$

$A = \{ \{1,2\}, \{3,4\}, \{5,6\} \}$
0 1 2

Q> Print matrix row by row.

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 6 | 7 | 8 | 9 | 10 |
| 2 | 11 | 12 | 13 | 14 | 15 |
| 3 | 16 | 17 | 18 | 19 | 20 |

⟶

1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20

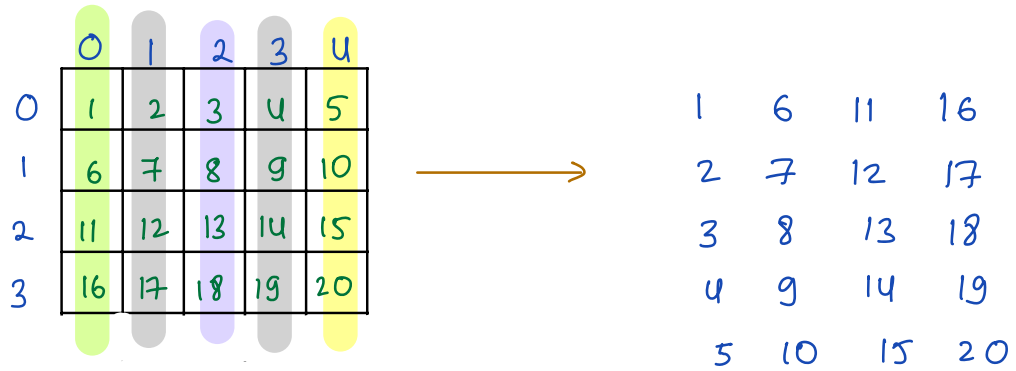
```
void printRowWise(int[][] A) {
    int R = A.length;      // 4
    int C = A[0].length;   // 5

    for (int r=0; r<R; r++) {   // 0..... 3
        for (int c=0; c<C; c++) {   // 0.... 4
            S.O.p(A[r][c] + " ");
        }
        sopln();
    }
}
```

| r | c |
|---|---|
| 0 | 0..... 4 |
| 1 | 0.... 4 |
| 2 | 0.... 4 |
| 3 | 0..... 4 |

1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20

Q> Print matrix column by column




```
void printColwise(int[][] A) {
    int R = A.length;    // 4
    int C = A[0].length; // 5

    for(int c=0; c<C; c++) {
        for(int r=0; r<R; r++) {
            S.O.P(A[r][c] + " ");
        }
        SOPln();
    }
}
```

Practice Matrix coding practice in IDE.

Java code for printing matrix row by row and column by column

How to take matrix as input ?

```java
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int R = scanner.nextInt();
    int C = scanner.nextInt();
    int[][] A = new int[R][C];

    for (int r = 0; r < R; r++) {
        for (int c = 0; c < C; c++) {
            A[r][c] = scanner.nextInt();
        }
    }

    for (int r = 0; r < R; r++) {
        for (int c = 0; c < C; c++) {
            System.out.print(A[r][c] + " ");
        }
        System.out.println();
    }
}
```

Break : 22:49

Q) Sum of Matrix

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 6 | 7 | 8 | 9 | 10 |
| 2 | 11 | 12 | 13 | 14 | 15 |
| 3 | 16 | 17 | 18 | 19 | 20 |

Sum = 210

```
void printRowWise(int[][] A) {
    int R = A.length;    // 4
    int C = A[0].length; // 5
    int sum = 0;
    for (int r=0; r<R; r++) {   // 0 ..... 3
        for (int c=0; c<C; c++) { // 0 .... 4
            sum += A[r][c];
        }
    }
    sop(sum);
}
```

Q) waveform Printing.

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 6 | 7 | 8 | 9 | 10 |
| 2 | 11 | 12 | 13 | 14 | 15 |
| 3 | 16 | 17 | 18 | 19 | 20 |

1 2 3 4 5
10 9 8 7 6
11 12 13 14 15
20 19 18 17 16

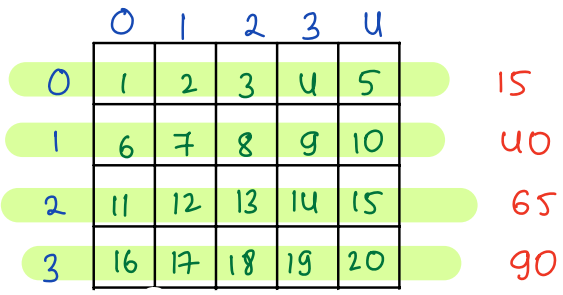
0 → 3 2
1 ← 7 1

3 2
1 7

for every even indexed row ⟶ left to right
for every odd indexed row ⟵ right to left

```
void waveform (int[][] A) {
    int R = A.length;    // 4
    int C = A[0].length; // 5
    for (int r=0 ; r<R ; r++) {  // 0 ..... 3
        if (r%2 ==0) {// even
            for (int c=0; c<C; c++) {
                S.O.p( A[r][c] + " ");
            }
        } else {
            for (int c = C-1; c>=0; c--) {
                sop (A[r][c] + " ");
            }
        }
        sopln ();
    }
}
```

Q> Row-wise sum




```
void printRowWise(int[][] A) {
    int R = A.length;      // 4
    int C = A[0].length;   // 5

    for (int r=0; r<R; r++) {   // 0 ..... 3
        int rowSum = 0;
        for (int c=0; c<C; c++) {   // 0 .... 4
            rowSum += A[r][c];
        }
        sopln(rowSum);
    }
}
```

| r | rowSum | c | rowSum | print |
|---|---|---|---|---|
| 0 | 0 | 0 ..... 4 | 15 | |
| 1 | 0 | 0 .... 4 | 40 | |
| 2 | 0 | 0 .... 4 | 65 | |
| 3 | 0 | 0 .... 4 | 90 | |

Q> Column wise max

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 1 | 3 | -2 | 7 |
| 1 | 9 | 0 | 8 | -1 |
| 2 | 5 | 6 | -2 | 3 |
| | 9 | 6 | 8 | 7 |

Try 가기

23:20

int[][] A = { {1, 3, -2, 7}, {9, 0, 8, -1}, {5, 6, -2, 3} }

```java
public class Main {
  public static void main(String[] args) {
      int[][] A = {{1, 3, -2, 7},
                   {9, 0, 8, -1},
                   {5, 6, -2, 3}};

      int R = A.length;
      int C = A[0].length;

      // To iterate column wise
      for(int c = 0; c < C; c++){
        int max = A[0][c];
        for(int r = 0; r < R; r++){
          if(max < A[r][c]){
            max = A[r][c];
          }
        }
        System.out.println(max);
      }
  }
}
```

Doubt Session

→ 100% PSP
→ 95% attendance
→ All live contest with reattempt
→ Mock interview cleared.