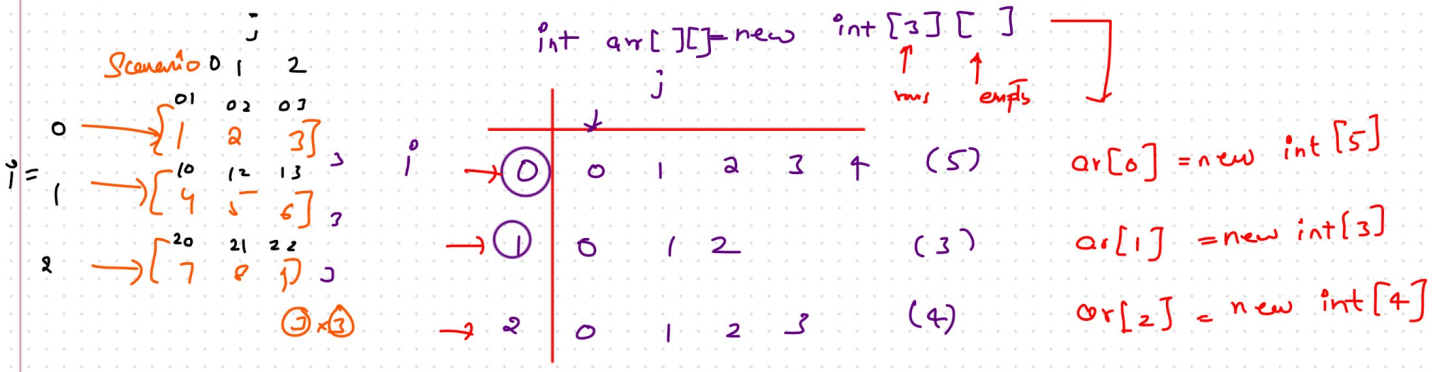
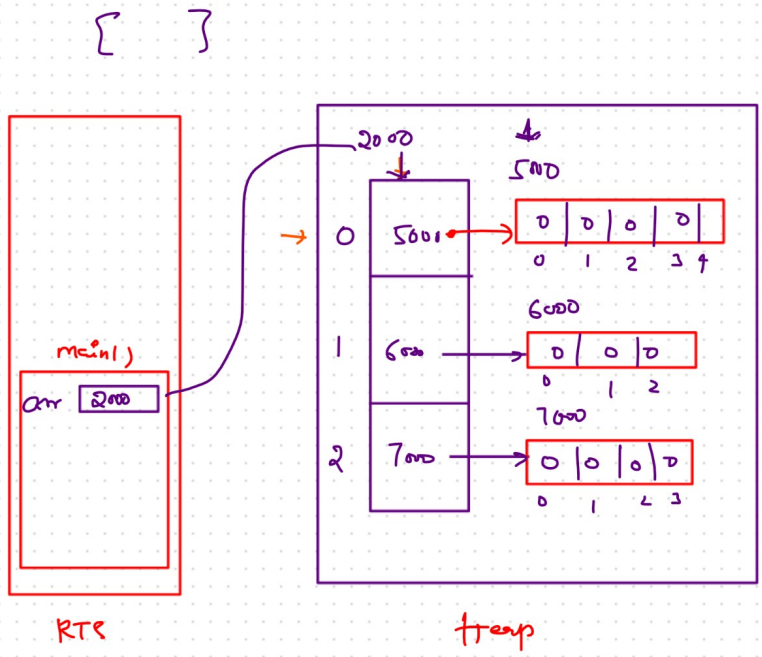
Jagged Array

Array with diff columns in each row



Memory Mapping

```
public static void main(String[] args) {
    int arr[][]=  new int[3][];
    arr[0] = new int[5];
    arr[1] = new int[3];
    arr[2] = new int[4];
```



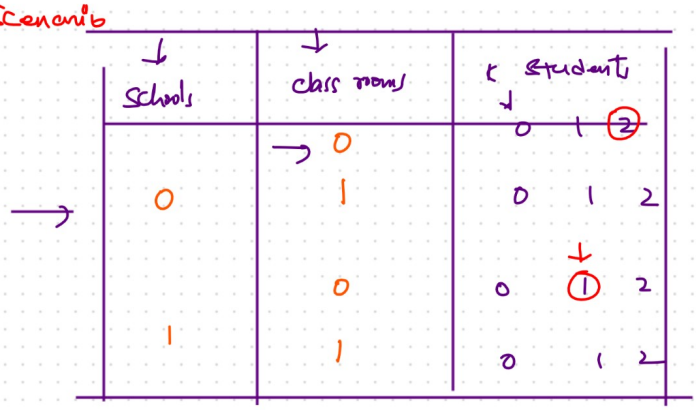
arr[0][1] = 0

Note:

Before	After
arr[0] → null	arr[0] → 5000
arr[1] → null	arr[1] → 6000
arr[2] → null	arr[2] → 7000

3D Array

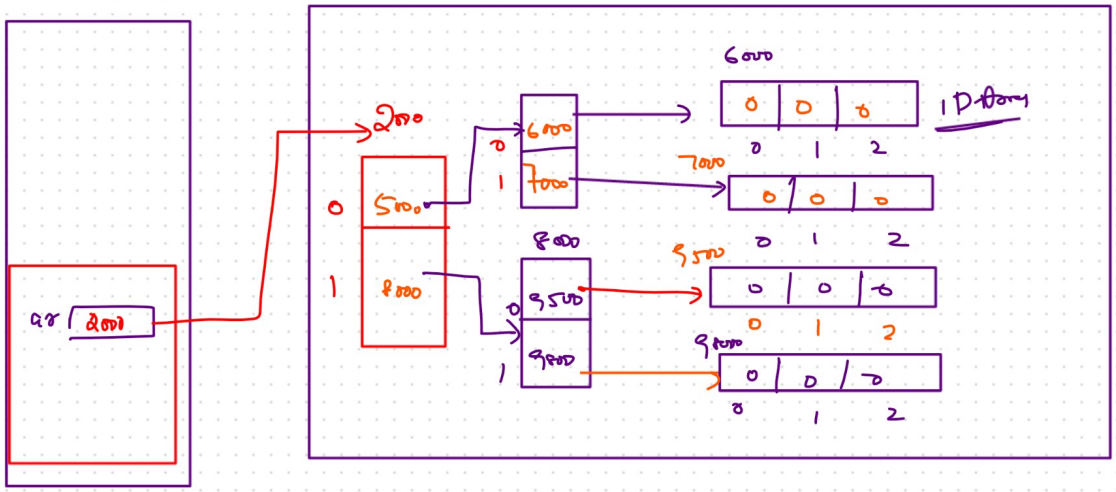
Scenario



school [0] [0] [2]

school [1] [0] [1]

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



3D - Jagged Array

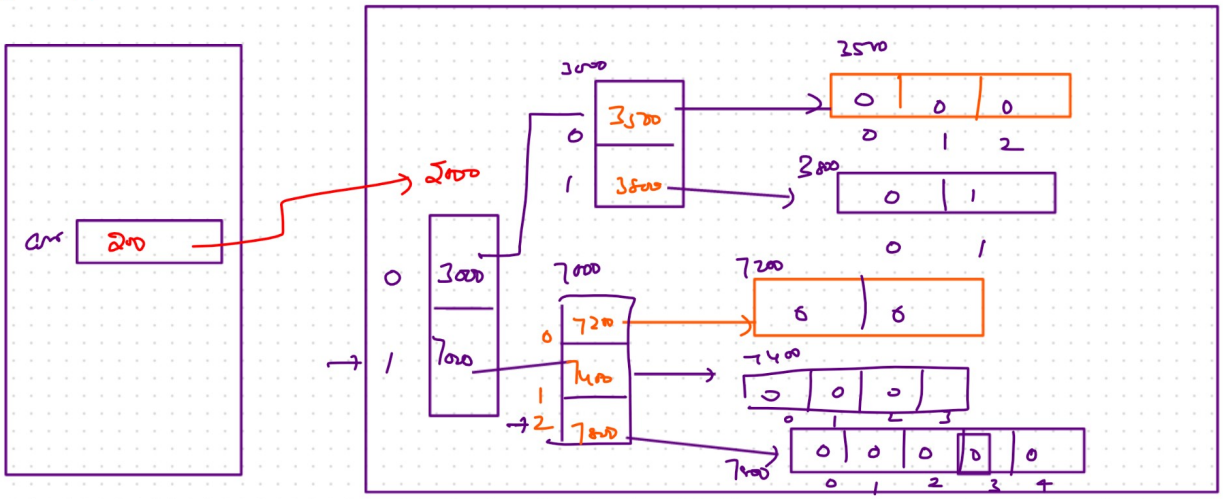
```
int ar[][][] = new int[2][][]
```

```
ar[0] = new [2][]
ar[1] = new [3][]
```

```
ar[0][0] = new int[3]
ar[0][1] = new int[2]
```

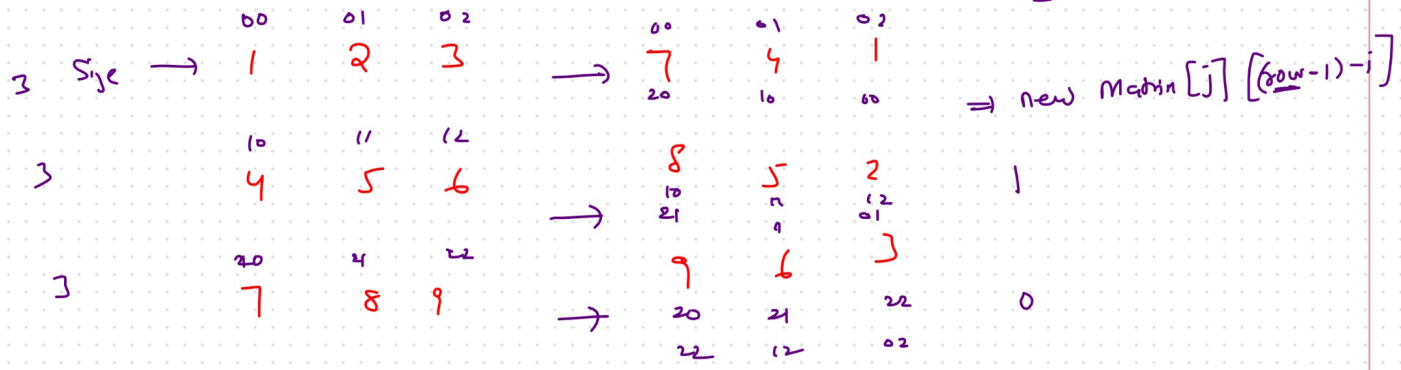
```
ar[1][0] = new int[2];
ar[1][1] = new int[4];
ar[1][2] = new int[5];
```

Memory mapping



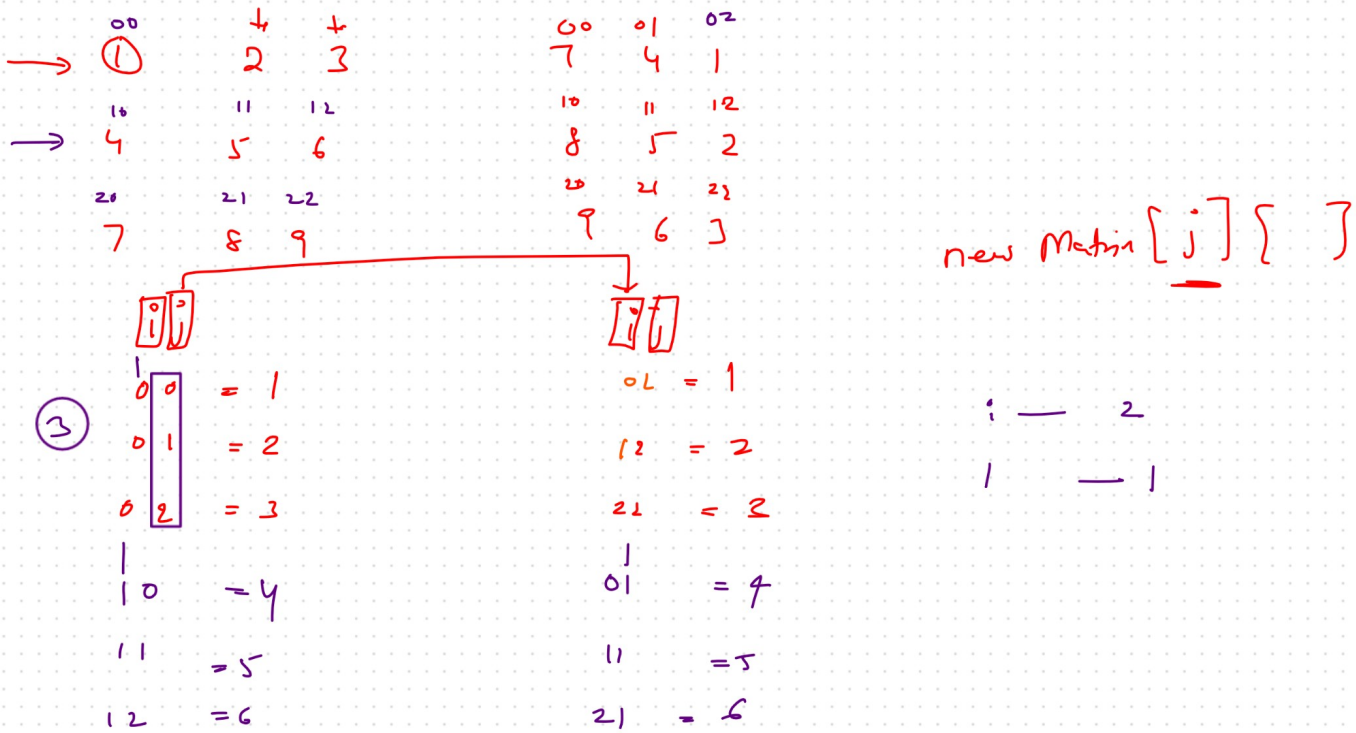
arr[1][2][3] = 0

Transpose Matrix 90°>



H.W

<90° new Matrix[(row-1)-i][j]



20 = 7

21 = 8

22 = 9

00 = 7

16 = 8

20 = 9

→

i	j
0	2
0	2
0	2
1	1
1	1
1	1
2	0
2	0
2	0

→ (3)

(row - 1) → (1)

i = 0

(3 - 1) - 0 = 2

i = 1

(row - 1) - i

(3 - 1 - 1) = 1

i = 2

(row - 1) - i

(3 - 1) - 2) = 0

Declaration & Initialization of Arrays

```
int ar[] = { 1, 2, 3, 4, 5, 6 }
```

```
int ar[3][2] = { { 1, 1 }, { 2, 2 }, { 3, 3 } }
                   0        1        2
```

```
int ar[][][] = { { { 5, 3, 1 }, { 2, 4 } }, { { 1, 1, 1 }
                 0                                { 2, 2, 2 }, { 3, 3 } } }
```

Alternative ways of declaring array

H.W

```
int ar[]
int[] ar;
int []ar;
```

```
int ar[][]
int [][]ar;
int[][] ar;
int []ar[];
int [] ar[];
int [] []ar;
```

mcq