⇒ 2D array

$$n\left\{\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 6 & 7 & 8 & 9 & 10 \\ 2 & 11 & 12 & \textcircled{13} & 14 & 15 \\ 3 & 16 & 17 & 18 & 19 & 20 \\ 4 & 21 & 22 & 23 & 24 & 25 \end{array}\right.$$

$\overbrace{\quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad}^{m}$

$\boxed{n}$ int $\textcircled{[]}$ arr = new int[n]

$\downarrow$

n = 5 , m = 5

int [][] arr = new int[n] [m];

access :-

↳ arr[2][2]

finding length :

n = arr.length; // no. of rows

m = arr[0].length; // no. of cols.

Print the Matrix Row-wise

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
  → int m = scn.nextInt();   // row no
    int n = scn.nextInt();   // col no
    int[][] arr = new int[m][n];
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }

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

Operations $= m * n$

T.C $= O(m * n)$

S.C $= O(1)$

Print Alternate Row

$$\begin{array}{cc|c|c|c|c|c|}
 & & 0 & 1 & 2 & 3 & 4 \\
\hline
\checkmark\checkmark & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
 & 1 & 6 & 7 & 8 & 9 & 10 \\
\hline
\checkmark\checkmark & 2 & 11 & 12 & \textcircled{13} & 14 & 15 \\
\hline
 & 3 & 16 & 17 & 18 & 19 & 20 \\
\hline
\checkmark\checkmark & 4 & 21 & 22 & 23 & 24 & 25 \\
\hline
\end{array}$$

code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int m = scn.nextInt();  // row no
    int n = scn.nextInt();  // col no
    int[][] arr = new int[m][n];
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }

    printAlternateRows(arr, m, n);
}

public static void printAlternateRows(int[][] arr, int m, int n) {
    for (int i = 0; i < m; i += 2) {   // rows
        for (int j = 0; j < n; j++) {   // cols
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
}
```

also valid

```
public static void printAlternateRows(int[][] arr, int m, int n) {
    for (int i = 0; i < m; i++) {   // rows
        if (i % 2 == 0) {
            for (int j = 0; j < n; j++) {   // cols
                System.out.print(arr[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Print Upper triangular matrix 1

	j = 0	j = 1	j = 2	j = 3	j = 4
i = 0	1	2	3	4	5
i = 1	6	7	8	9	10
i = 2	11	12	13	14	15
i = 3	16	17	18	19	20
i = 4	21	22	23	24	25

which values to print

$i = 0, \; j = 0, 1, 2, 3, 4$

$i = 1, \; j = 1, \underline{2, 3, 4}$

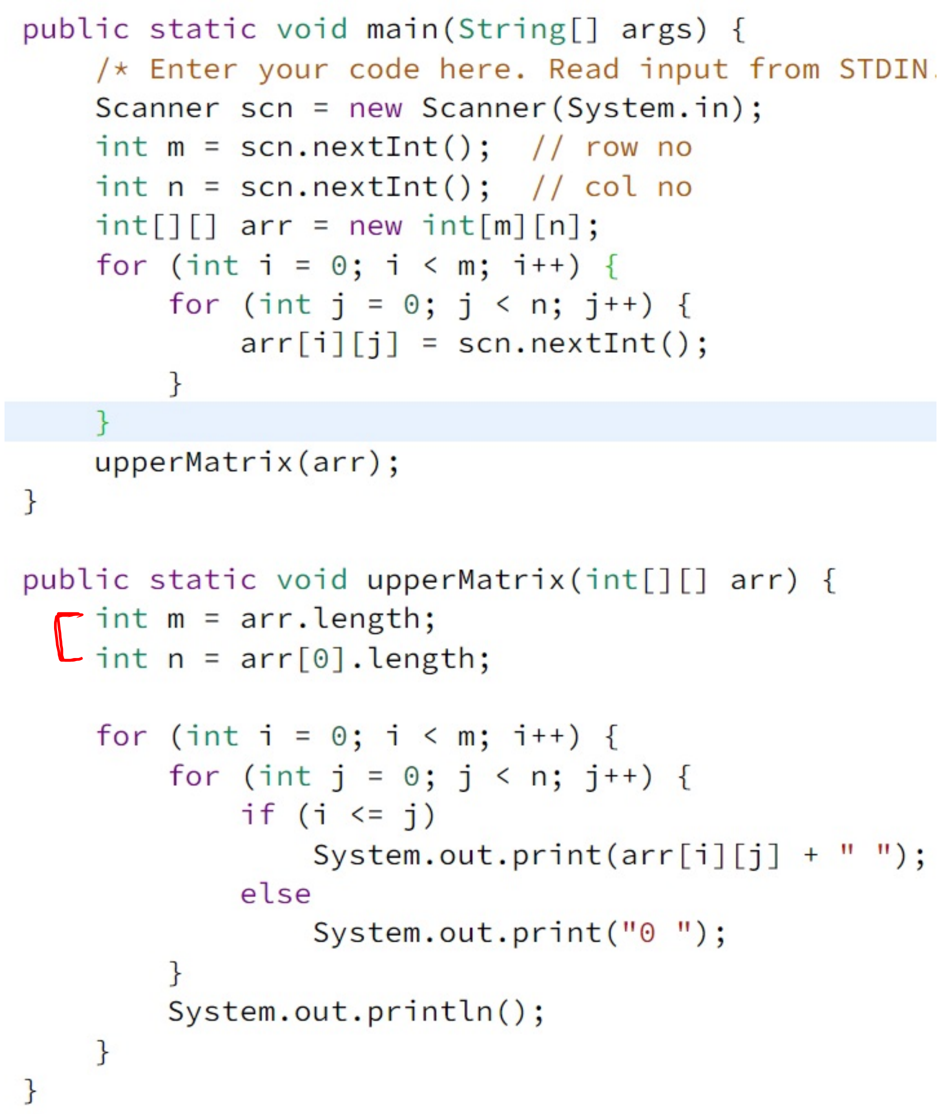
$i = 2, \; j = 2, 3, 4$

$i = 3, \; j = 3, 4$

$i = 4, \; j = 4$

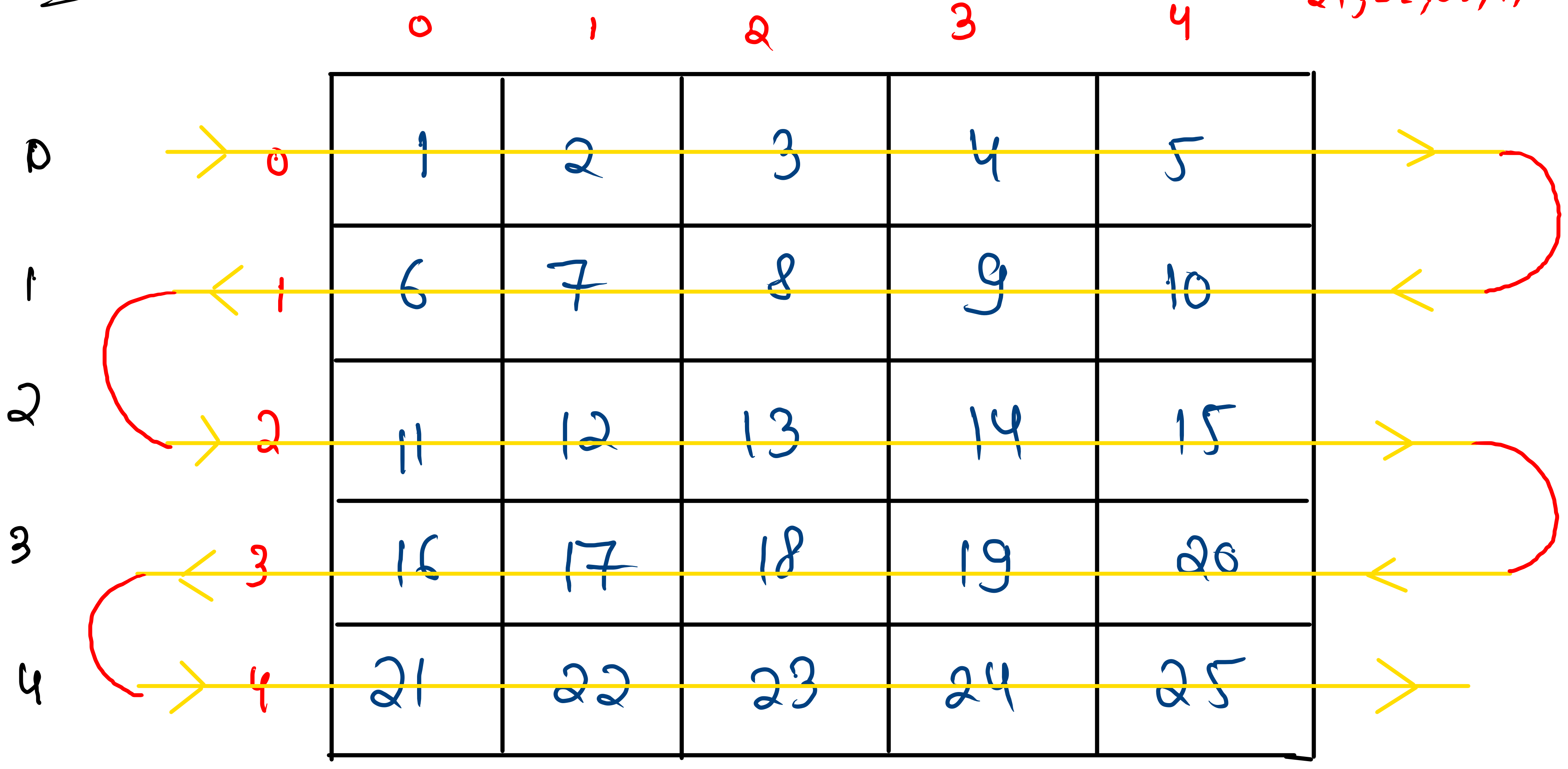
condition $\underline{i <= j}$

code



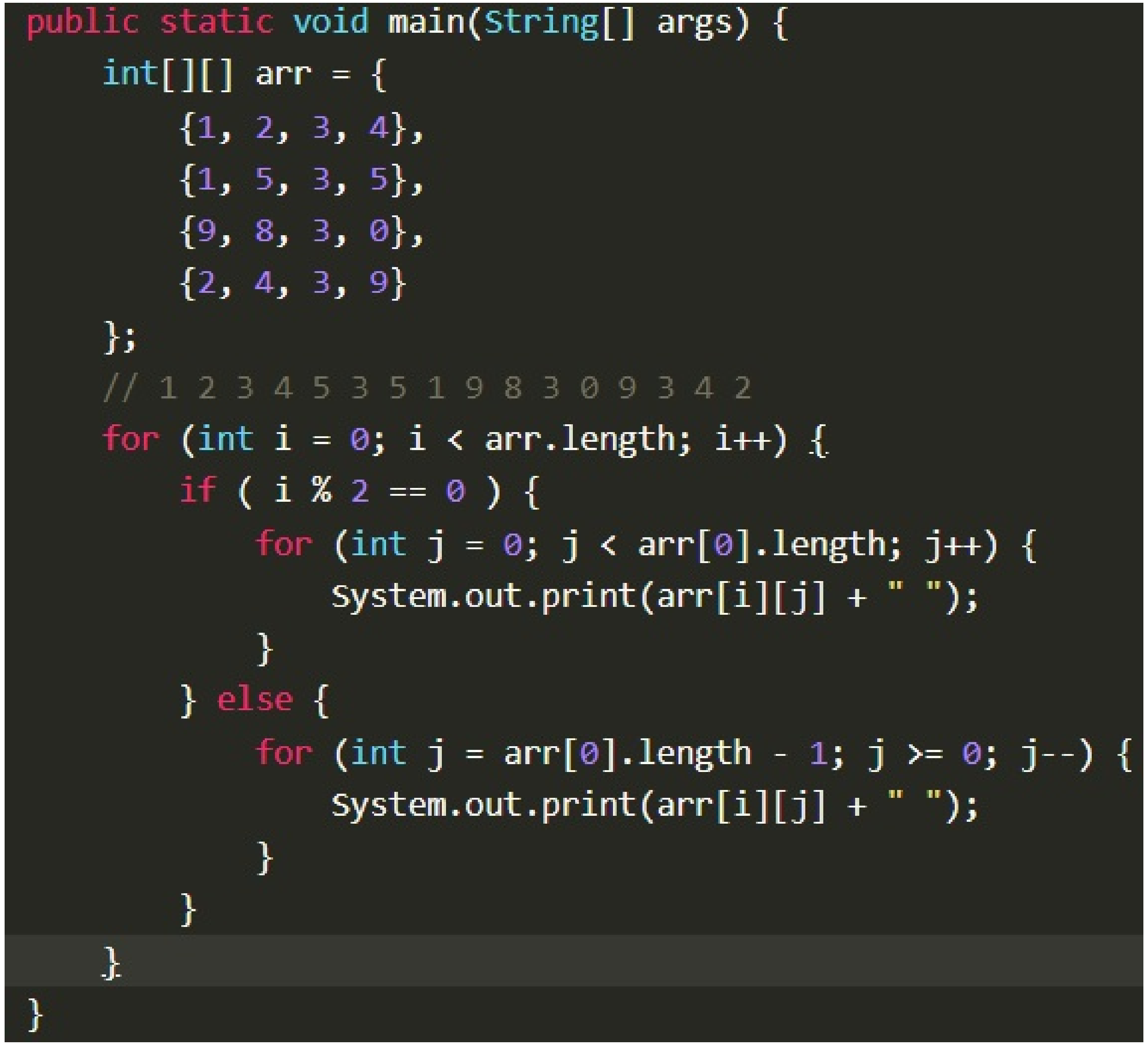
```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN
    Scanner scn = new Scanner(System.in);
    int m = scn.nextInt();  // row no
    int n = scn.nextInt();  // col no
    int[][] arr = new int[m][n];
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }
    upperMatrix(arr);
}

public static void upperMatrix(int[][] arr) {
    int m = arr.length;
    int n = arr[0].length;

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

Ques print:- 1, 2, 3, 4, 5, 10, 9, 8, 7, 6, 11, 12, 13, 14, 15, 20, 19, 18, 17, 16, 21, 22, 23, 24, 25



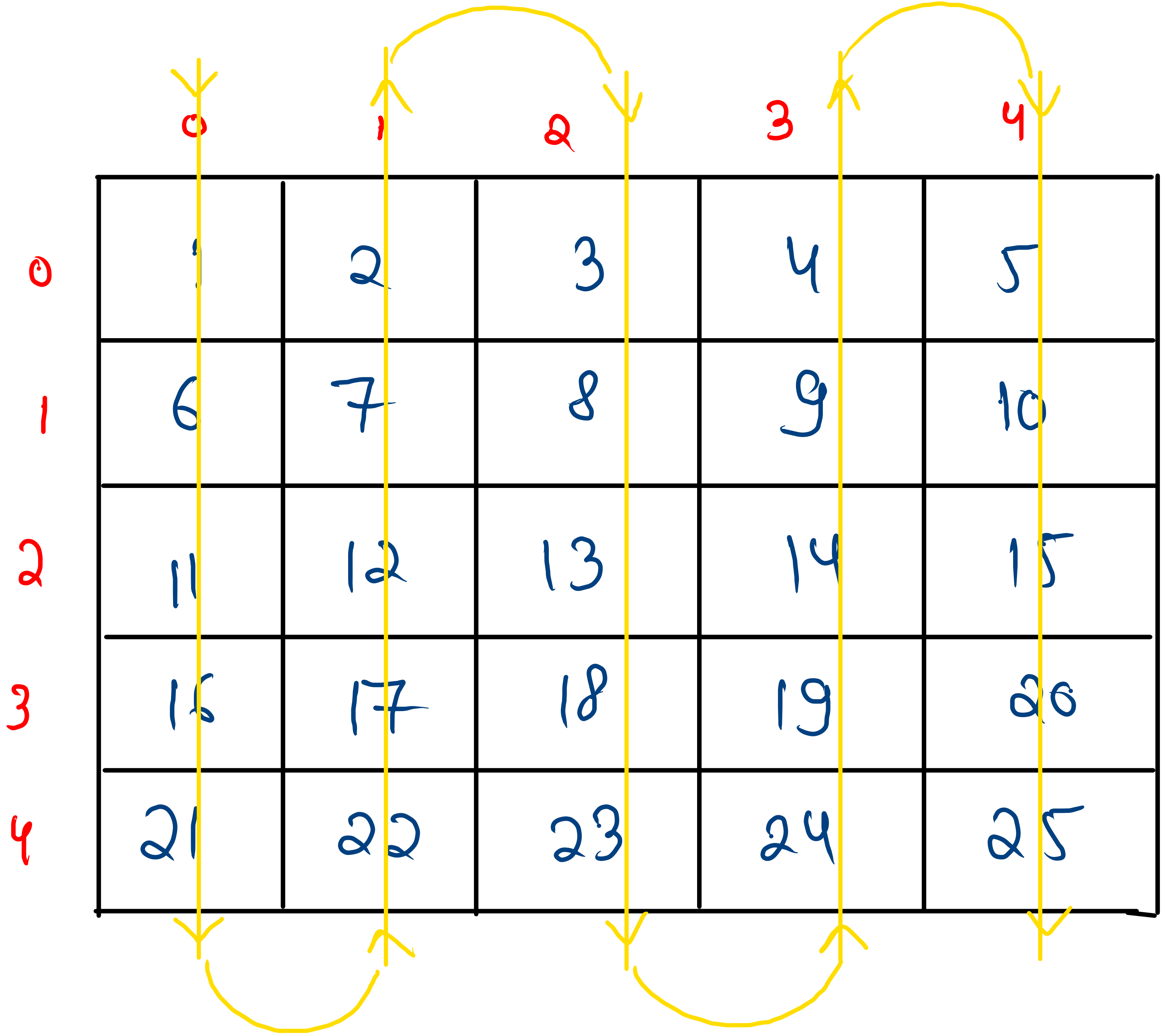
$$
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 1 & 2 & 3 & 4 & 5 \\
1 & 6 & 7 & 8 & 9 & 10 \\
2 & 11 & 12 & 13 & 14 & 15 \\
3 & 16 & 17 & 18 & 19 & 20 \\
4 & 21 & 22 & 23 & 24 & 25
\end{array}
$$

Code



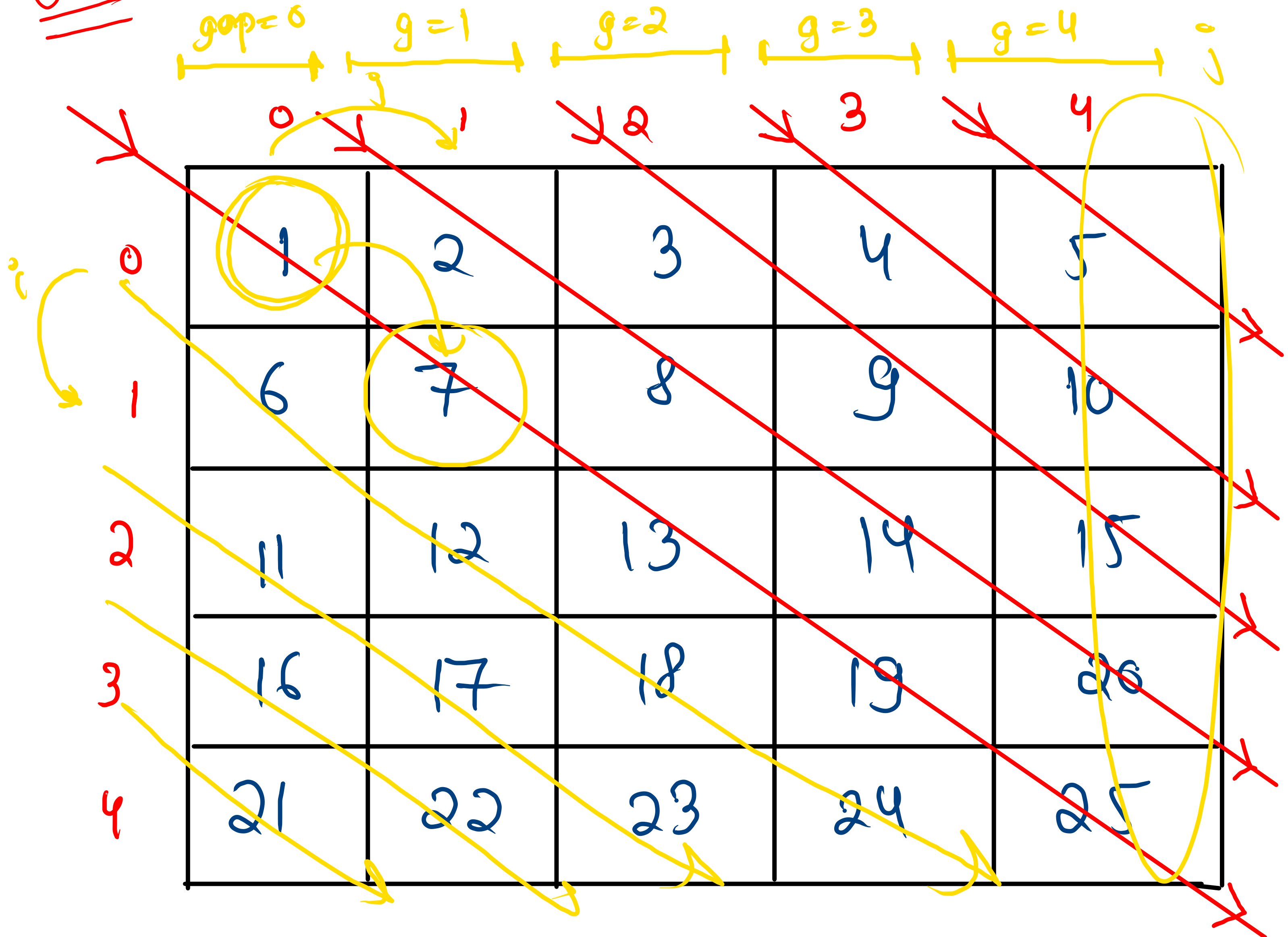
```java
public static void main(String[] args) {
    int[][] arr = {
        {1, 2, 3, 4},
        {1, 5, 3, 5},
        {9, 8, 3, 0},
        {2, 4, 3, 9}
    };
    // 1 2 3 4 5 3 5 1 9 8 3 0 9 3 4 2
    for (int i = 0; i < arr.length; i++) {
        if ( i % 2 == 0 ) {
            for (int j = 0; j < arr[0].length; j++) {
                System.out.print(arr[i][j] + " ");
            }
        } else {
            for (int j = arr[0].length - 1; j >= 0; j--) {
                System.out.print(arr[i][j] + " ");
            }
        }
    }
}
```

Ques)



$$\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 1 & 2 & 3 & 4 & 5 \\
1 & 6 & 7 & 8 & 9 & 10 \\
2 & 11 & 12 & 13 & 14 & 15 \\
3 & 16 & 17 & 18 & 19 & 20 \\
4 & 21 & 22 & 23 & 24 & 25
\end{array}$$

Que) print :- 1, 7, 13, 19, 25, 2, 8, 14, 20, 3, 9, 15, 4, 10, 5

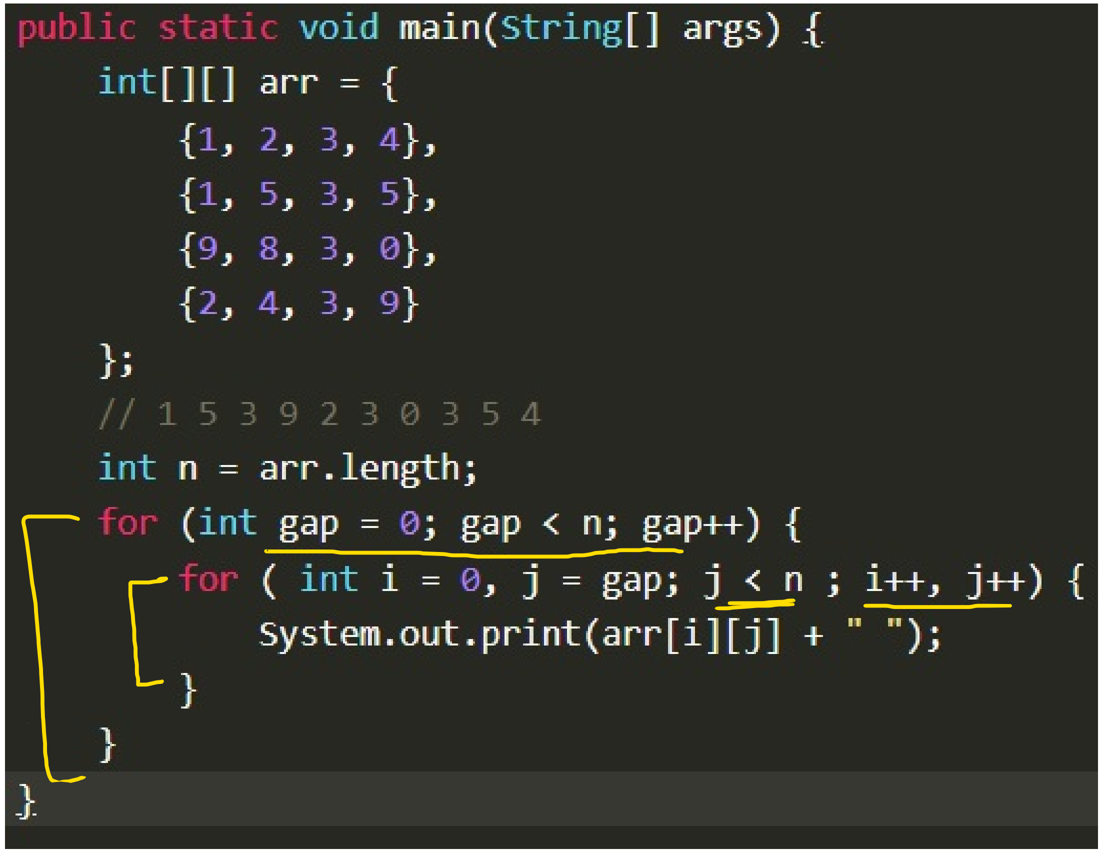


Que)

what is the starting point,

when to stop,

how to move



```java
public static void main(String[] args) {
    int[][] arr = {
        {1, 2, 3, 4},
        {1, 5, 3, 5},
        {9, 8, 3, 0},
        {2, 4, 3, 9}
    };
    // 1 5 3 9 2 3 0 3 5 4
    int n = arr.length;
    for (int gap = 0; gap < n; gap++) {
        for ( int i = 0, j = gap; j < n ; i++, j++) {
            System.out.print(arr[i][j] + " ");
        }
    }
}
```

$\boxed{gap \rightarrow 0 \rightarrow ij}$

for (i=0, j=0,1,2,3,4 , $\bigcirc$ j<n , i++, j++)

i j

0,0

0,1

0,2

0,3

0,4