Good String Checker

str = "abacdbcd"

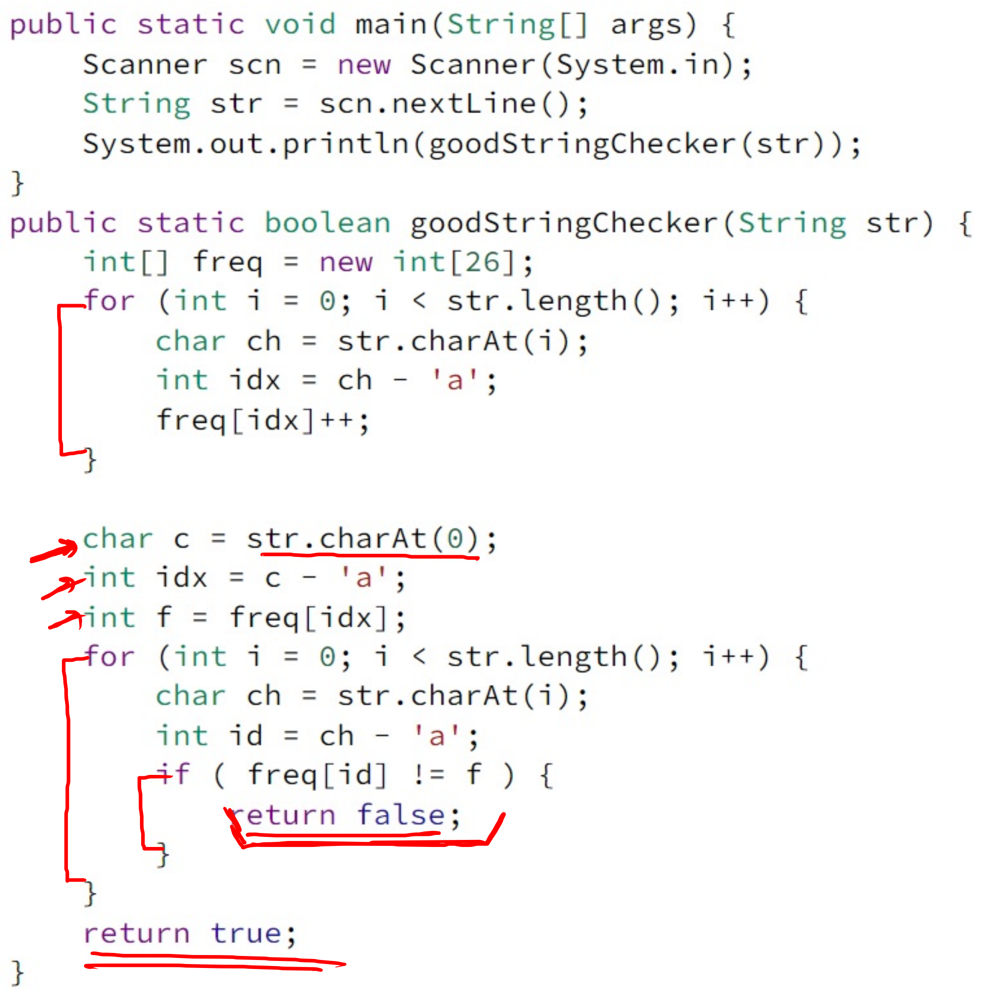
freq =
(int)

0	1	2	3	4	5	
2	2	2	2	0	0	....

psudo code

```
1) create freq array of size 26
2) traverse in string
    2.1) increase freq of each character
3) traverse in string
    3.1) check if any freq is diff.
        3.1.1) return false
4) return true
```

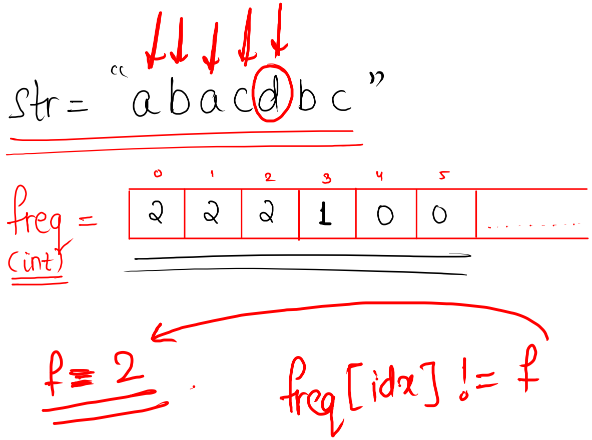
code



```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    System.out.println(goodStringChecker(str));
}
public static boolean goodStringChecker(String str) {
    int[] freq = new int[26];
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - 'a';
        freq[idx]++;
    }

    char c = str.charAt(0);
    int idx = c - 'a';
    int f = freq[idx];
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int id = ch - 'a';
        if ( freq[id] != f ) {
            return false;
        }
    }
    return true;
}
```

T.C=O(N)
where N is
size of string

S.C= O(1)



Str = "abacdbc"

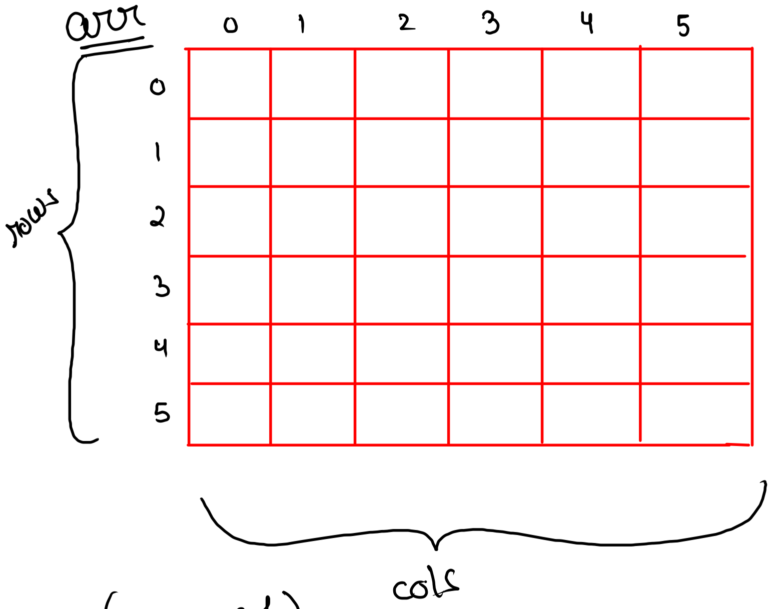
freq = (int)

0	1	2	3	4	5
2	2	2	1	0	0

f = 2

freq[idx] != f

⇒ 2D arrays



(row, col)

no. of rows = arr.length ;

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

declare

```
1D :-  int[] arr = new int[size];
2D :-  int[][] arr = new int [row size] [col size];
```

access each index

```
int a = arr[3];        // single loop
int b = arr[3][2];     // nested loop
```

$\uparrow$ row index $\quad \uparrow$ col index

update

```
arr[2] = 5;
arr[2][3] = 6;
```

Print the Matrix Row-wise

i/p)

m = 3 // no. of rows

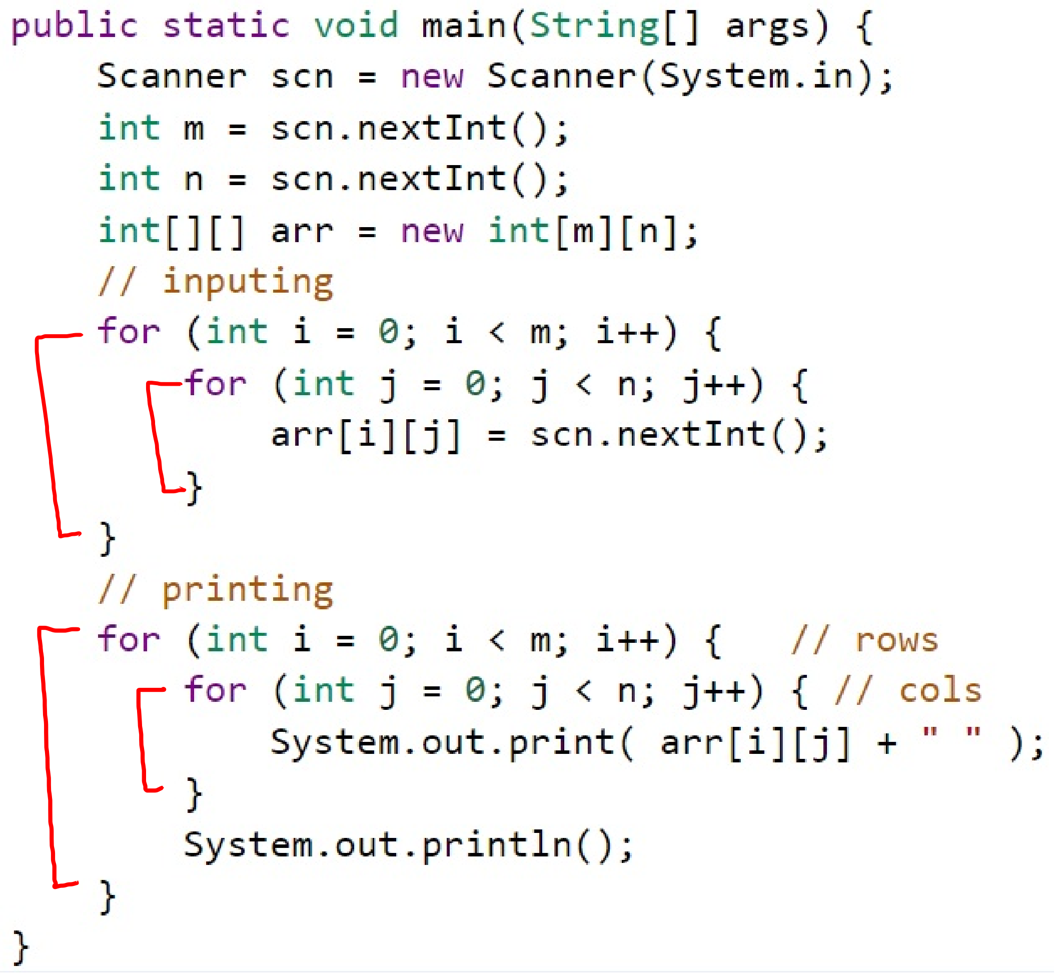
n = 4 // no. of cols

arr =

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12

```
for ( int i=0; i< m; i++) {
    for (int j=0; j<n; j++) {
        arr[i][j]= scn.nextInt();
    }
}
```

code



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int m = scn.nextInt();
    int n = scn.nextInt();
    int[][] arr = new int[m][n];
    // inputing
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }
    // printing
    for (int i = 0; i < m; i++) {    // rows
        for (int j = 0; j < n; j++) { // cols
            System.out.print( arr[i][j] + " " );
        }
        System.out.println();
    }
}
```

T.C = O(m * n)

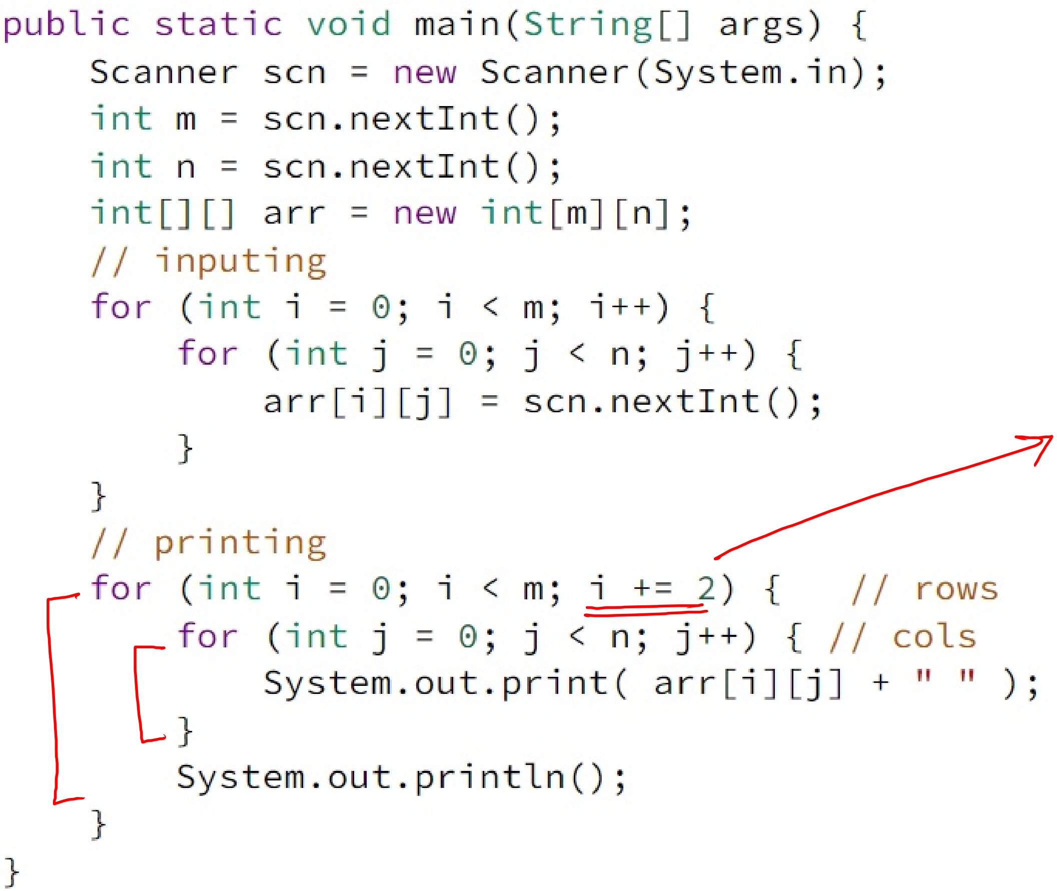
m = size of rows

n = size of cols

(linear)

S.C = O(m * n)

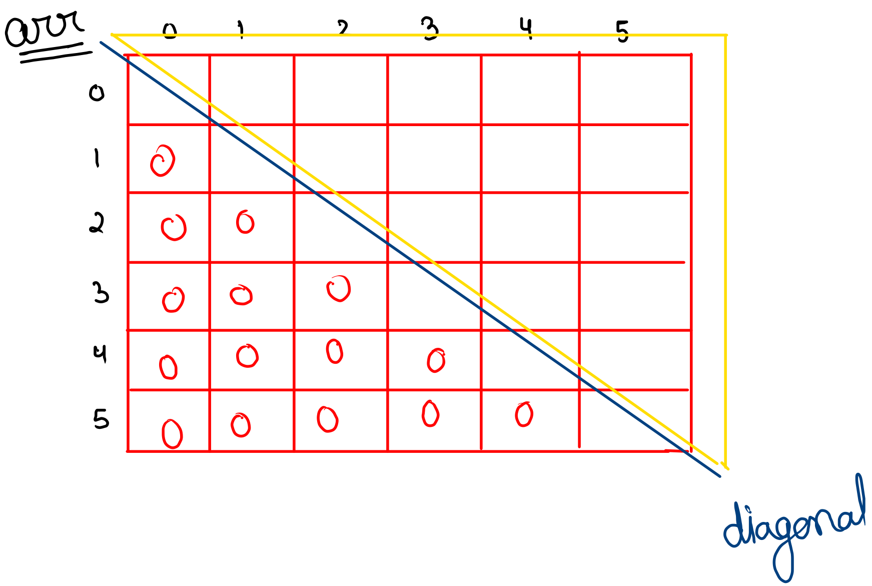
Print Alternate Row



```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int m = scn.nextInt();
    int n = scn.nextInt();
    int[][] arr = new int[m][n];
    // inputing
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }
    // printing
    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();
    }
}
```

only change

Print Upper triangular matrix 1



$$\begin{array}{c|cccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
0 & & & & & & \\
1 & 0 & & & & & \\
2 & 0 & 0 & & & & \\
3 & 0 & 0 & 0 & & & \\
4 & 0 & 0 & 0 & 0 & & \\
5 & 0 & 0 & 0 & 0 & 0 &
\end{array}$$

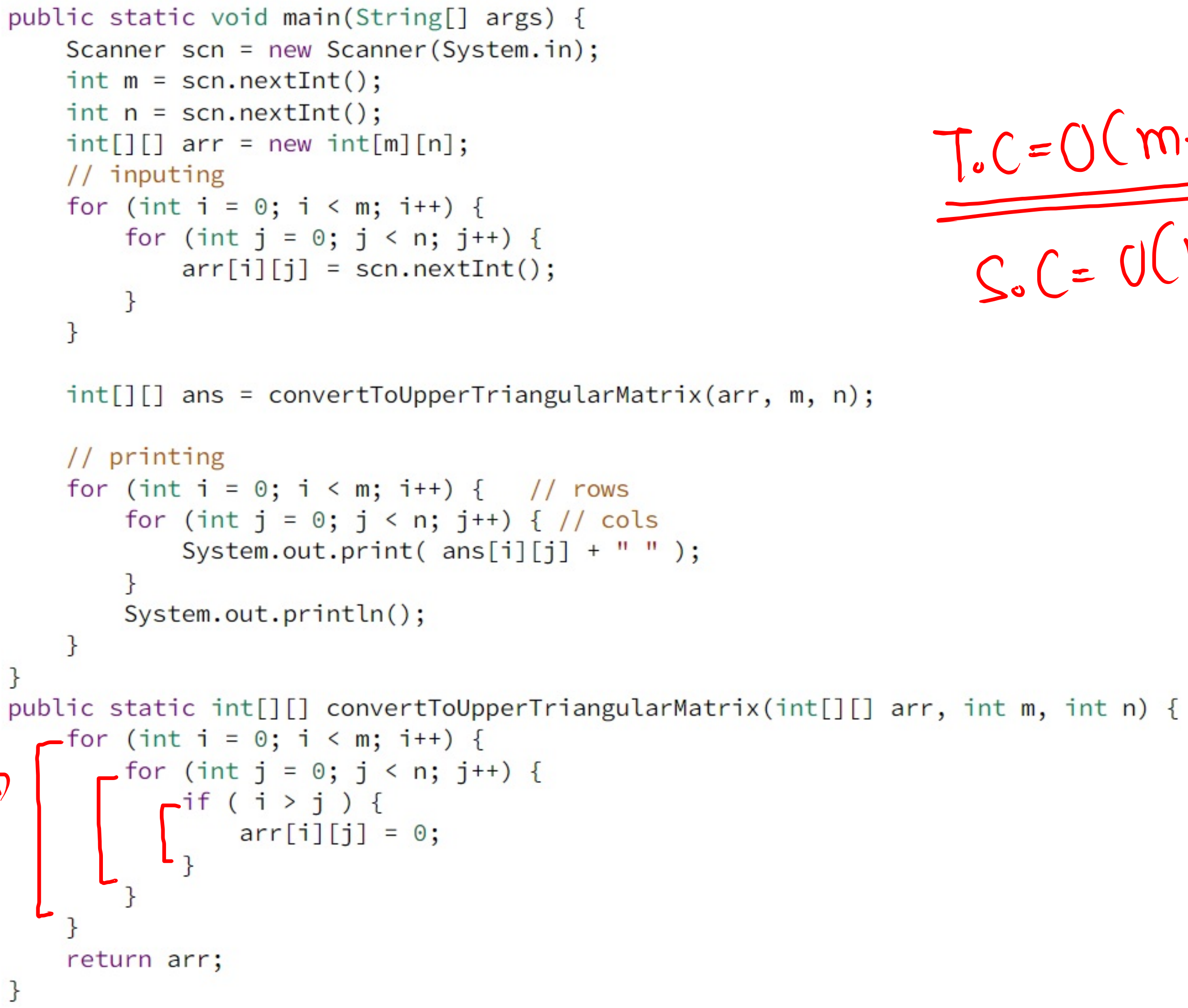
(indexing)

(row, cols)

(1, 0)
(2, 0)
(2, 1)
(3, 0)
(3, 1)
(3, 2)
(4, 0)
(4, 1)
(4, 2)
(4, 3)
(5, 0)
(5, 1)
(5, 2)
(5, 3)
(5, 4)

$$\left(\begin{array}{c} row > col \\ i > j \end{array} \right)$$

code



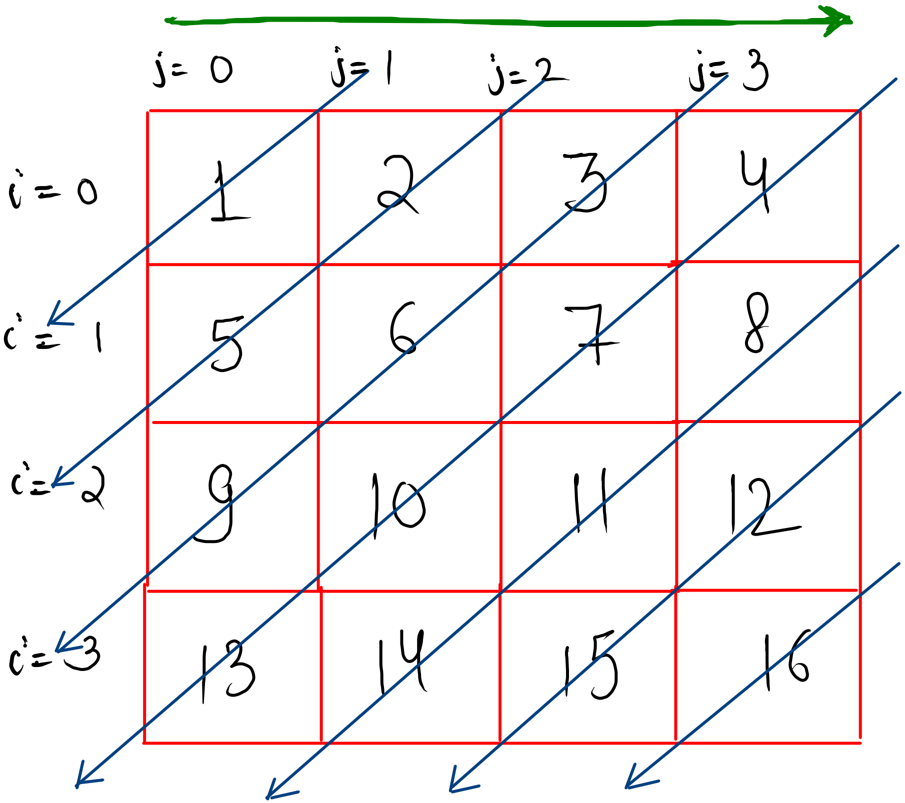
```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int m = scn.nextInt();
    int n = scn.nextInt();
    int[][] arr = new int[m][n];
    // inputing
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }

    int[][] ans = convertToUpperTriangularMatrix(arr, m, n);

    // printing
    for (int i = 0; i < m; i++) {   // rows
        for (int j = 0; j < n; j++) { // cols
            System.out.print( ans[i][j] + " " );
        }
        System.out.println();
    }
}
public static int[][] convertToUpperTriangularMatrix(int[][] arr, int m, int n) {
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            if ( i > j ) {
                arr[i][j] = 0;
            }
        }
    }
    return arr;
}
```

T.C=O(m*n)

S.C= O(m*n)

upgradation

Print the matrix left-diagonal wise



	j= 0	j= 1	j= 2	j= 3
i = 0	1	2	3	4
i = 1	5	6	7	8
i = 2	9	10	11	12
i = 3	13	14	15	16

ans :- 1, 2, 5, 3, 6, 9, 4, 7, 10, 13, 8, 11, 14, 12, 15, 16

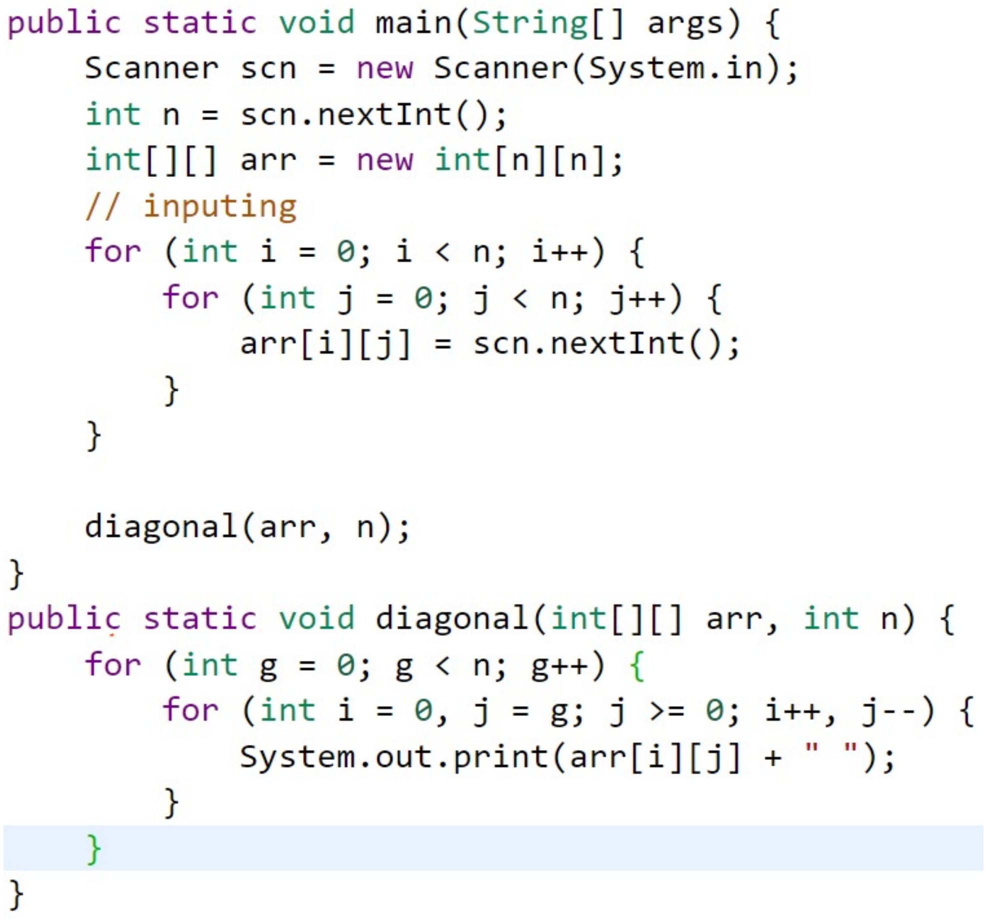
starting	stopping cond^n	upgradation
(0,0)	j>=0	i++
(0,1)		j--
(0,2)		
(0,3)		
row, col		

i row = 0

j col = 0, 1, 2, 3

```
for( int g=0; g<n ; g++){
    for( int i=0, j=g ; j>=0; i++, j--){
        Syso( arr[i][j]+" ");
    }
}
```

Code



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[][] arr = new int[n][n];
    // inputing
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }

    diagonal(arr, n);
}
public static void diagonal(int[][] arr, int n) {
    for (int g = 0; g < n; g++) {
        for (int i = 0, j = g; j >= 0; i++, j--) {
            System.out.print(arr[i][j] + " ");
        }
    }
}
```