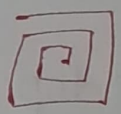
2-D arrays

Creation-

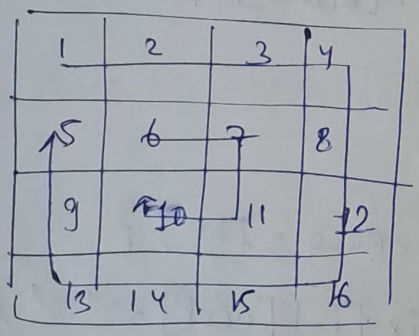
```
int matrix[][] = new int[size1][size2];
```

Spiral Matrix



amazon, google, apple, oracle

eg:



output:

1 2 3 4 8 12 16 15 14 13

9 5 6 7 11 10

Approch -

iteration :→	1st	2nd	matlab
variable ↓ startRow =	0	1	++
startCol =	0	1	++
endRow =	n-1 (3)	2	--
endCol =	n-1 (3)	2	--

// Code in LAPTOP

PS

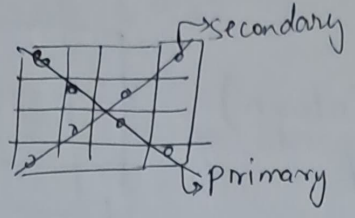
```
void printSpiral(int matrix[][]) {
    int sRow = 0, sCol = 0;
    int eRow = eCol = matrix.length - 1;

    while (sRow <= eRow && sCol <= eCol) {
        // top
        for (int j = sCol; j <= eCol; j++) {
            System.out.print(matrix[sRow][j] + " ");
        }
        // right
        for (int i = sRow + 1; i <= eRow; i++) {
            System.out.print(matrix[i][eCol] + " ");
        }
        // bottom
        for (int j = eCol - 1; j >= sCol; j--) {
            System.out.print(matrix[eRow][j] + " ");
            if (sRow == eRow) break;
        }
        // left
        for (int i = eRow - 1; i >= sRow + 1; i--) {
            if (sCol == eCol) break;
            System.out.print(matrix[i][sCol] + " ");
        }
        sCol++; sRow++; eRow--; eCol--;
    }
```

Diagonal Sum

Concept :.

$O(n)$

Sum:



Sum of ele of primary array =

condition => i == j

Sum += arr[i][j] where i == j

~~Sow~~ so we can replace j

Sum += arr[i][i]

Sum of ele of secondary array =

condition: If (i + j = n - 1)

Then j = n - 1 - i

Therefore,

Sum += arr[i][n-1-i]

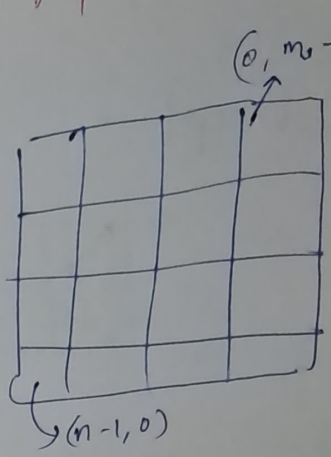
Code -

```
ps ~~voi~~ int diagonalSum(int matrix[][]) {
for (int i=0; i<matrix.length; i++) {
    sum += matrix[i][i];
    if (i != matrix.length-1-i)
        sum += matrix[i][matrix.length-1-i];
} ~~return~~
return sum;
}
```

Search in Sorted Matrix

(1) Brute force $O(n^2)$
Search everywhere

(2) Row wise / Column wise $O(n \log n)$
Search in each R/C seperately



(3) Staircase Approach $O(m+n)$

$(n-1, 0)$	$(0, m-1)$
key < cell value	key < cell value
TOP	LEFT
key > cell value	key > cell value
RIGHT	BOTTOM

CODE- ps ~~void~~ boolean staircaseSearch (int matrix[][], int key)

```
{ int row=0, col = matrix.length-1;
  while (row < matrix.length && col >=0) {
    if (matrix[row][col] == key) {
      System.out.print(" ____ ");
      return true;
    }
    else if (key < matrix[row][col])
      col--;
    else row++;
}
```