# DAY 1

## INTERVIEW BIT PROBLEMS :

### Spiral Order Matrix I

Given a matrix of m * n elements (m rows, n columns), return all elements of the matrix in spiral order.
Example:
Given the following matrix:

```
[
    [ 1, 2, 3 ],
    [ 4, 5, 6 ],
    [ 7, 8, 9 ]
]
```

You should return
[1, 2, 3, 6, 9, 8, 7, 4, 5]

**CODE :**

***C***

```
/**
 * @input A : Read only ( DON'T MODIFY ) 2D integer array
 * @input n11 : Integer array's ( A ) rows
 * @input n12 : Integer array's ( A ) columns
 *
 * @Output Integer array. You need to malloc memory, and fill the length in len1
 */
int* spiralOrder(const int** A, int n11, int n12, int *len1) {
    *len1=n11*n12;
    int *output_list=(int *) malloc(*len1 * sizeof(int));
//declaring output list
    int r1=0,c1=0,r2=n11,c2=n12;
```

```
    /*r1-starting row , r2=ending row ,c1=starting column ,
c2=ending column*/
    int ind=0;
    int i;
    while(r1<r2 && c1<c2){
        //printing the first row elements
        for(i=c1;i<c2;++i){
            output_list[ind++]=A[r1][i];
        }
        r1++;
        //printing the last column elements
        for(i=r1;i<r2;++i){
            output_list[ind++]=A[i][c2-1];
        }
        c2--;
        if(r1<r2){
            //printing the last row
            for(i=c2-1;i>=c1;--i){
                output_list[ind++]=A[r2-1][i];
            }
            r2--;
        }
        if(c1<c2){
            //printing the first column
            for(i=r2-1;i>=r1;--i){
                output_list[ind++]=A[i][c1];
            }
            c1++;
        }
    }
    return output_list;
}
```

**PYTHON**

```
class Solution:
    # @param A : tuple of list of integers
    # @return a list of integers
    def spiralOrder(self, A):
        r1=0
        r2=len(A)
        c1=0
        c2=len(A[0])
        output_list=[]
        while r1<r2 and c1<c2:
            #printing the first row
            for i in range(c1,c2):
                output_list.append(A[r1][i])
            r1+=1
            #printing the last column
            for i in range(r1,r2):
                output_list.append(A[i][c2-1])
            c2-=1
            #printing the last row
            for i in range(c2-1,c1-1,-1):
                output_list.append(A[r2-1][i])
            r2-=1
            #printing the first column
            for i in range(r2-1,r1-1,-1):
                output_list.append(A[i][c1])
            c1+=1
        return output_list
```