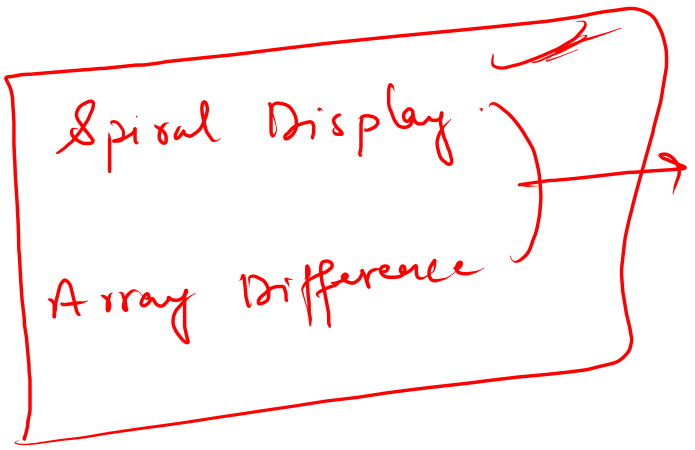
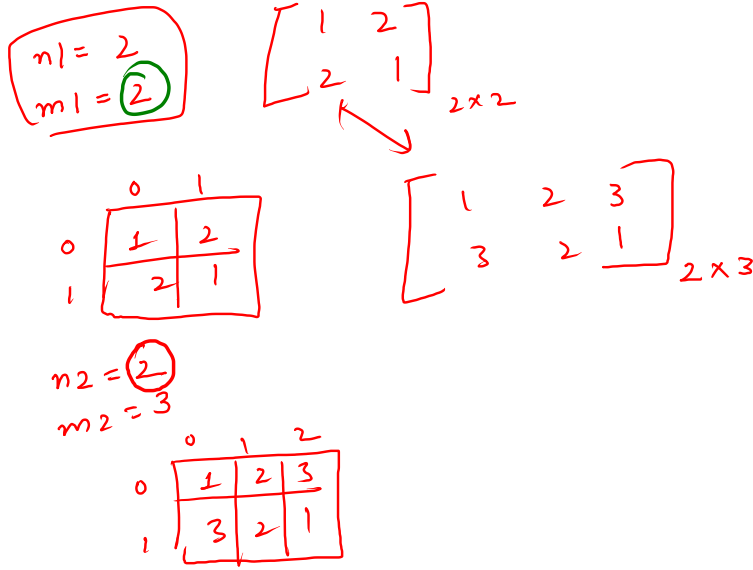
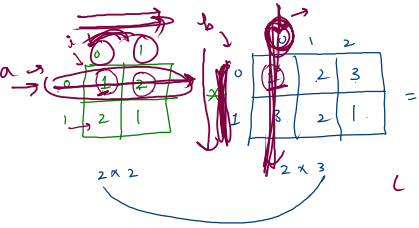

Spiral Display
Array Difference

Matrix Multiplication (16 July)

Problem | Submissions | Leaderboard | Discussions

1. You are given a number n1, representing the number of rows of 1st matrix.
2. You are given a number m1, representing the number of columns of 1st matrix.
3. You are given n1*m1 numbers, representing elements of 2d array a1.
4. You are given a number n2, representing the number of rows of 2nd matrix.
5. You are given a number m2, representing the number of columns of 2nd matrix.
6. You are given n2*m2 numbers, representing elements of 2d array a2.
7. If the two arrays representing two matrices of dimensions n1 * m1 and n2 * m2 can be multiplied, display the contents of prd array as specified in output Format.
8. If the two arrays can't be multiplied, print "Invalid input".





	0	1	2
0	1*1 + 2*3 = 1+6 = 7	1*2 + 2*2 = 2+4 = 6	1*3 + 2*1 = 3+2 = 5
1	2*1 + 1*3 = 2+3 = 5	2*2 + 1*2 = 4+2 = 6	2*3 + 1*1 = 6+1 = 7

a[i][0] * b[0][0] + a[i][1] * b[1][0]

1*1 + 2*3 = 1+6 = 7

a[i][?] * b[?][j]

→ i → Considering a particular row.

→ j → Considering a particular col.

→ k → Traversing over the i^{th} row & j^{th} col.

m1 * m2

m1 = 2

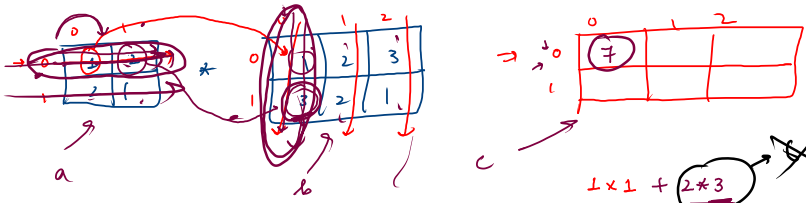
m2 = 3

m1, m2

matrix 1 → row wise

matrix 2 → column wise traversal.

```
for(int i=0; i<m1; i++)
{
  for(int j=0; j<m2; j++)
  {
    for(int k=0; k<m1; k++) {
      c[i][j] += a[i][k] * b[k][j];
    }
```



Multiply ith row with jth col.

$C[0][0] = a[0][0] * b[0][0]$

$C[0][0] = a[0][1] * b[1][0]$

$i = 0 < 2$

$j = 0 < 3$

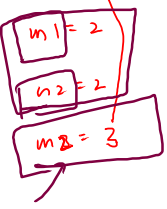
$k = 0 < 2$

$k = 1$

$C[0][0] = a[0][0] * a[0][0]$

$C[0][0] = 1x1 + = a[0][1] * b[1][0]$

$C[0][0] = 1*1 + 2x3$



0^{th} row * 0^{th} col. ⇒ $a[0][-] * b[-][0] *$

* 1^{st} col

* 2^{nd} col.

$C[row][col] = a[row][k] * b[k][col]$

Traversing a particular row & a particular col.

```
for(int i=0; i<n1; i++){
  for(int j=0; j<m2; j++){
    for(int k=0; k<n2/m1; k++)
      c[i][j] += a[i][k] * b[k][j];
  }
}
```

Complete the code

Day Rem

10 min

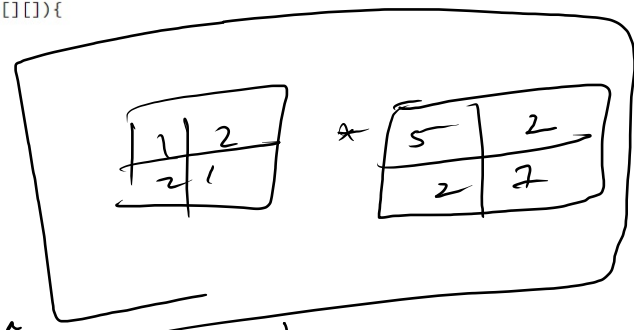
```
public static void matrixMultiplication(int a[][], int b[][]){
    int n1 = a.length;
    int m1 = a[0].length;

    int n2 = b.length;
    int m2 = b[0].length;


    if(m1 != n2){
        System.out.println("Invalid input");
    }else{
        int c[][] = new int[n1][m2];

        for(int i=0;i<n1;i++){
            for(int j=0;j<m2;j++){
                for(int k=0;k<n2;k++){
                    c[i][j] += a[i][k]*b[k][j];
                }
            }
        }

        for(int i=0;i<n1;i++){
            for(int j=0;j<m2;j++){
                System.out.print(c[i][j] +" ");
            }
            System.out.println();
        }
    }
}
```



Convert 1-D Array to 2-D Array (16 July)

Problem | Submissions | Leaderboard | Discussions

Take an array of size N as input, representing a 1-D array. There are many possible factors of N, for eg:- p * q = N. Now take p and q as input and print the 2-D array with dimensions as p*q.

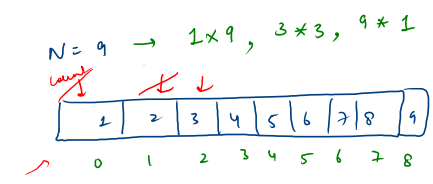
Note: It is guaranteed that a 2-D array will be formed.

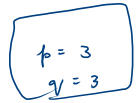
Sample Input 0

```
9
1 2 3 4 5 6 7 8 9
3 3
```

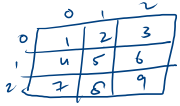
Sample Output 0

```
1 2 3
4 5 6
7 8 9
```



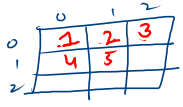



```
int count = 0;
for(int i=0; i<p; i++){
  for(int j=0; j<q; j++){
    arr[i][j] = a[count];
    count++;
  }
}
```



→ 1D array → 2D array p*q

arr →



count = 0 + 2 = 3

i = 0 < 3

j = 0 < 3 (T) ✗

arr[0][0] = a[0]

arr[0][1] = a[1]

arr[0][2] = a[2]

i = 1 < 3 (T)

j = 0 < 3 (T)

arr[1][0] = a[3]

arr[1][1] = a[4]

```java
public class Solution {
    public static void convert1DTo2D(int a[],int N,int p,int q){
        int arr[][] = new int[p][q];

        int count =0;
        for(int i=0;i<p;i++){
            for(int j=0;j<q;j++){
                arr[i][j] = a[count];
                count++;
            }
        }

        for(int i=0;i<p;i++){
            for(int j=0;j<q;j++){
                System.out.print(arr[i][j] + " ");
            }
            System.out.println();
        }
    }
```

→ 5 min

+

5 min → Shift Matrix Reverse

Shift Matrix Row-Wise (21 july)

Problem | Submissions | Leaderboard | Discussions

Given a n*n matrix and an integer k. Shift the matrix elements row-wise by k. Print the final matrix such that all elements of the row in one line.

Sample Input 0

```
3
9 0 5
2 7 5
2 3 3
2
```

Sample Output 0

```
9 0 5
5 2 7
3 2 3
```

k=2 ⇒ 1

k-1

k=2

n=3

Steps:-

reverse the given row

n=5

k=1

k=2

k=k-1

reverse the row

0, k-2

k-1, n-1

k=2

k-1=2-1=1

k=2-1 ⇒ k=k-1

(0, n-1)

(0, k-2)

(k-1, n-1)

k=2-1=1

k=2-2) =0

(0, 0)

(1, 4)

k

In general

reverse the any

(0, k-1)

(k, n-1)


```java
public static void reverse(int arr[][],int row, int left,int right){
    while(left<right){
        int temp = arr[row][left];
        arr[row][left] = arr[row][right];
        arr[row][right] = temp;

        left++;
        right--;
    }
}

public static void shiftRowwise(int arr[][], int n, int k){
    for(int i=0;i<n;i++){
        reverse(arr,i,0,n-1);
        reverse(arr,i,0,k-1);
        reverse(arr,i,k,n-1);
    }

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

```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN.
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int arr[][] = new int[n][n];
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            arr[i][j] = scn.nextInt();
        }
    }

    int k = scn.nextInt();
    k = k-1;

    shiftRowwise(arr,n,k);
}
```

Modify The Matrix (21 july)

Problem | Submissions | Leaderboard | Discussions

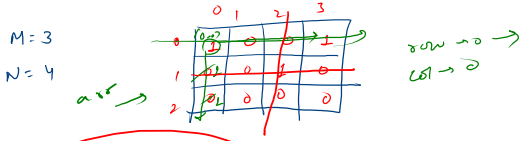
Given a boolean matrix mat[M][N] of size M X N, modify it such that if a matrix cell mat[i][j] is 1 (or true) then make all the cells of ith row and jth column as 1.

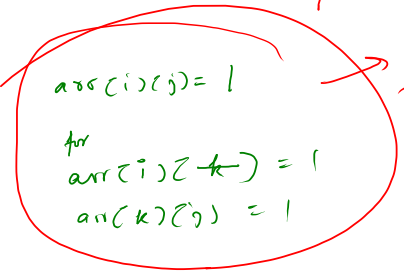
Sample Input 0

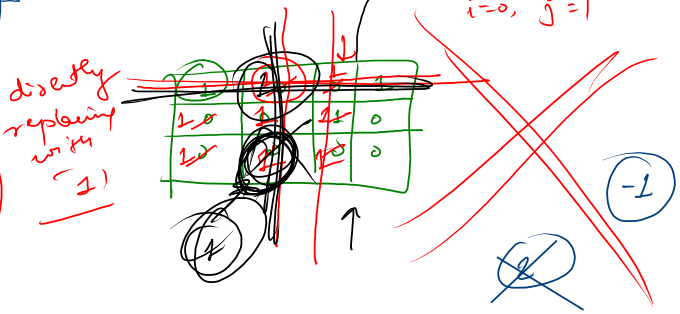
```
3
4
1 0 0 1
0 0 1 0
0 0 0 0
```

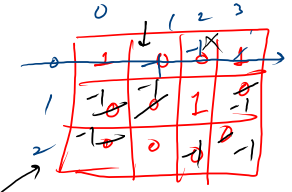
Sample Output 0

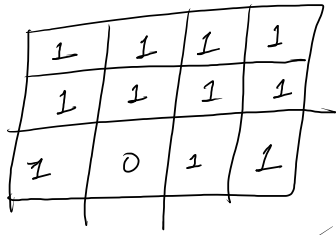
```
1 1 1 1
1 1 1 1
1 0 1 1
```











```java
public static void fillRowAndColWithMinusOne(int arr[][],int row,int col){
    for(int j=0;j<arr[0].length;j++){ // For filling row with value -1
        if(arr[row][j] !=1){
            arr[row][j] = -1;
        }
    }

    for(int i =0;i<arr.length;i++){ // For filling column with value -1
        if(arr[i][col] !=1){
            arr[i][col] = -1;
        }
    }

}
```

```java
public static void modifyTheMatrix(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(arr[i][j] == 1){
                fillRowAndColWithMinusOne(arr,i,j);
            }
        }
    }

    for(int i=0;i<m;i++){
        for(int j=0;j<n;j++){
            if(arr[i][j] == -1){
                arr[i][j]=1;
            }
        }
    }


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


}
```