10/5/25

2D ARRAY:

Consist of rows and coloumns.

Syntax

int[][]arr=new int[1][3]

very first square bracket represent rows second represent coloumn

- while declaring the array we have to declare number of rows and coloumn numbers are optional.
- Each row will act as an indiviual array

Size and length:

arr[5]={1,2,3};

size is 5: maximum capacity

length is 3: will be actual capacity

- (learn address mapping)

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int[][]arr=new int[3][2];
        for(int i = 0;i<arr.length;i++){
            for(int j=0;j<arr[i].length;j++){
                arr[i][j] = in.nextInt();
            }
        }
        for(int i = 0;i<arr.length;i++){
            for(int j=0;j<arr[i].length;j++){
                System.out.println(arr[i][j]);

            }
            System.out.println();
        }
```

```
    }
}
```

Q. Check wheather the given array is sorted or not

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        int[]arr={3,4,7,2,1};

        boolean isSorted=true;
        for(int i=1;i<arr.length;i++){
            if(arr[i]<arr[i-1]){
                isSorted=false;
                break;
            }
        }

        if(isSorted){
            System.out.println("Array Sorted");
        }
        else{
            System.out.println("Not Sorted");
        }

    }
}
```

Q. Check wheather the element is there are not

```
class Main {
    public static void main(String[] args) {
        int[]arr={10,20,30,40,50};
        int target = 30;

        boolean found=false;
        for(int i=0;i<arr.length;i++){
            if(arr[i]==target){
                found = true;
                break;
            }
```

```
        }
        if(found){
            System.out.println("Element "+ target + " is Present");
        }
        else{
            System.out.println("Element "+ target + " is not Present");

        }

    }
}
```

Q. [1,0,1,0,1,0] - - > [0,0,0,1,1,1] without sorting

```
class Main {
    public static void main(String[] args) {
        int[] arr = {1,0,1,0,1,0};
        int n = arr.length;
        int i= 0;
        int j =n-1;

        while(i<j){
            if(arr[i]==1){
                i++;
            }

            if(arr[j]==0){
                j--;
            }
            if (i < j && arr[i] == 0 && arr[j] == 1){
                int temp = arr[i];
                arr[i] = arr[j];
                i++;
                j--;
            }

        }
        System.out.println("Array : ");
        for(int x: arr){
            System.out.print(x+" ");
        }
    }
}
```

Q. Swapping of two number without using third variable

```
class Main {
   public static void main(String[] args) {
      int a= 10;
      int b=20;

      System.out.println("Before swapping");
      System.out.println("a="+ a+ ",b=" + b);

      a=a+b;
      b=a-b;
      a=a-b;

      System.out.println("After swapping");
      System.out.println("a="+a+",b=" + b);

   }
}
```

using array

```
class Main {
   public static void main(String[] args) {
      int[] arr = {10, 20};

      System.out.println("Before swapping:");
      System.out.println("arr[0] = " + arr[0] + ", arr[1] = " + arr[1]);

      arr[0] = arr[0] + arr[1];
      arr[1] = arr[0] - arr[1];
      arr[0] = arr[0] - arr[1];

      System.out.println("After swapping:");
      System.out.println("arr[0] = " + arr[0] + ", arr[1] = " + arr[1]);
   }
}
```

Q. Sum of zig-zag elements - - > 1,2,3,5,7,8,9 - >35

```
class Main {
    public static void main(String[] args) {
        int[] arr= {1,2,3,5,7,8,9};
        int sum =0;

        for(int num:arr){
            sum +=num;
        }
        System.out.println("Zig-zag sum = " + sum);

    }
}
```

Q. Maximum Consecutive Ones

Q. 1 2 3 147 741

456 258 852

789 369 963