14/11/2024
ARIKALEESWARAN G
22AM002

1. Given a sorted array, arr[] containing only 0s and 1s, find the transition point, i.e., the first index where 1 was observed, and before that, only 0 was observed. If arr does not have any 1, return -1. If array does not have any 0, return 0.

   Program:

```
class Solution {
    int transitionPoint(int arr[]) {
        // code here
        for(int i=0;i<arr.length;i++){
            if(arr[i] == 1){
                return i;
            }
        }
        return -1;
    }
}
```

   TC: O(long n)
   SC: O(1)

2. Given an array arr[], find the first repeating element. The element should occur more than once and the index of its first occurrence should be the smallest.

Program:

```
class Solution {

    public static int firstRepeated(int[] arr) {
        HashMap<Integer,Integer> hm = new HashMap<>();
              int a = Integer.MAX_VALUE;
        for(int i =0;i<arr.length;i++){
            if(hm.containsKey(arr[i])){
                  a = Math.min(a, hm.get(arr[i]));
            }
            hm.put(arr[i] ,i);
        }
        if(a == Integer.MAX_VALUE) return -1;

        return a + 1;
    }
}
```

TC: O(n)
SC: O(n)

3. Given a sorted array arr. Return the size of the modified array which contains only distinct elements.

Program:

```
class Solution {
    public int remove_duplicate(List<Integer> arr) {

      int k=1;
      int c=0;
      for(int i=1;i<arr.size();i++){
         if(!arr.get(c).equals(arr.get(i))){
            c++;
            arr.set(c,arr.get(i));
            k++;
          }
      }
      return k;
   }
}
```

TC: O(n)
SC: O(1)