Question : First and Last Occurrences

code:

```
class GFG {

    ArrayList<Integer> find(int arr[], int x) {

        ArrayList<Integer> num = new ArrayList<>();


        for (int i = 0; i < arr.length; i++) {

            if (arr[i] == x && (i == 0 || arr[i-1] != x)) {

                num.add(i);

            }


            if (arr[i] == x && (i == arr.length - 1 || arr[i+1] != x)) {

                num.add(i);

            }

        }


        if (num.size() == 0) {

            num.add(-1);

            num.add(-1);

        }


        return num;

    }

}
```

Question : Coin Change (Count Ways)

code:

```
class Solution {

    public int count(int coins[], int sum) {

        int[] dp = new int[sum + 1];

        dp[0] = 1;


        for (int i = 0; i < coins.length; i++) {

            for (int j = coins[i]; j <= sum; j++) {

                dp[j] += dp[j - coins[i]];

            }

        }


        return dp[sum];

    }

}
```

Question : Find Transition Point

code:

```
class Solution {

    int transitionPoint(int arr[]) {

        for (int i = 0; i < arr.length; i++) {

            if (arr[i] == 1) {

                return i;

            }

        }

        return -1;
```

```
    }
}
```

Question : First Repeating Element

code:

```
class Solution {
    public static int firstRepeated(int[] arr) {
        HashMap<Integer , Integer> num = new HashMap<>();
        for(int i = 0 ; i < arr.length ; i++){
            if(num.containsKey(arr[i])){
                return num.get(arr[i]);
            }
            num.put(arr[i],i);
        }
        return -1;
    }
}
```

Question : Remove Duplicates Sorted Array

code:

```
class Solution {
    public static int removeDuplicates(List<Integer> arr) {
        if (arr.size() == 0) return 0;

        int j = 0;
        for (int i = 1; i < arr.size(); i++) {
```

```
            if (!arr.get(i).equals(arr.get(j))) {
                j++;
                arr.set(j, arr.get(i));
            }
        }
        return j + 1;
    }
}
```