Practice Question

Solution 1:-

```
class Solution {
    public int peakElement(int[] arr, int n) {
        int low = 0;
        int high = n - 1;

        while (low < high) {
            int mid = low + (high - low) / 2;

            if (arr[mid] > arr[mid + 1]) {
                high = mid; // Potential peak is on the left side
            } else {
                low = mid + 1; // Potential peak is on the right side
            }
        }

        return low; // 'low' represents a peak element
    }
}
```

Solution 2:-

```
class Solution {
    public long count(int coins[], int N, int sum) {
        long[] dp = new long[sum + 1];
        dp[0] = 1; // Base case

        for (int coin : coins) {
            for (int i = coin; i <= sum; i++) {
                dp[i] += dp[i - coin];
            }
        }
```

```
        return dp[sum];
    }
}
```