4] * First and Last Position of target

arr = [ 0:5, 1:7, 2:7, 3:7, 4:7, 5:8, 6:8, 7:10 ] target = 7

first ↑ (1)  last ↑ (4)

- we simply apply binary search for finding first occurance.

- if first occurence is found than we apply binary search for finding last occurence.

- if we found target than we assume that it may be potential answer,

and we continue looking a first and last occurance.

```
if (target == mid) {

    ans = mid;

    if (isFirst == true) {   // looking for first occurrence in left side
        end = mid - 1;
    } else {                 // looking for last occurrence in Right side
        start = mid + 1;
    }
```

- isFirst is a boolean variable that defines where to apply search for first position or last position.

Code:

```
Find First and Last Position (int[] arr, int target) {

    int[] ans = {-1, -1};
    // look for First Position
    int FirstPosition = binarySearch(arr, target, true)
    ans[0] = FirstPosition;
    if (FirstPosition != -1) {
        // if First position exist look for last positi
        ans[1] = binarySearch(arr, target, False);
    }

    return ans;

}

binarySearch (int[] arr, int target, boolean isFirst){
    // isFirst is for finding first or last
    int start = 0; ans = -1;          Position
    int end = arr.length -1;

    while (start <= end){
        int mid = start + (end - start) /2;

        if (target > arr[mid]) {
            start = mid+1;
        }
        else if ( target < arr[mid]) {
            end = mid-1;
        }
```

```
        else{
            ans = mid;
            // it may be potential answer
        but further check for first and
        last

            if(isFirst == true){
                end = mid-1;
            }else{
                start = midt1;
            }
        }

    }

    return ans;
}
```