LOWER BOUND & UPPER BOUND

☆ In this problem, we are supposed to return the lowest index after which every element is bigger or equal to our target. The exact opposite of this is the upper bound problem.

Using Linear Search, this can be done in less than O(N) time. But we can optimise it using a Binary Search. We assume that if our answer variable does not exist, the default ans is highest possible index. So we start with a low pointer, a high pointer and our answer as highest answer.

If the mid element is lesser than our target element, we move to the right. If it is greater than our target, we set it as our answer and move to the left

Pseudocode:

```
lowerBoundBinarySearch(arr, N, a) {
    low = 0, ans = N - 1
    high = N - 1
```

```
while (low <= high) {
    mid = (low + high) / 2
    if (arr[mid] == a) {
        while (arr[mid] == arr[mid-1]) {
            mid--
        }
        return mid
    } else if (arr[mid] < a) {
        low = mid + 1
    } else {
        ans = mid
        high = mid - 1
    }
}
return ans
}
```

The exact opposite of this is done to find Upper Bound.