* Questions on Binary Search algorithm:

I) Ceiling of Number:

- smallest element in array greater or equal(=) to target.

arr = $[\overset{0}{2}, \overset{1}{3}, \overset{2}{5}, \overset{3}{9}, \overset{4}{14}, \overset{5}{16}, \overset{6}{18}]$ target = 15

ceiling(arr, 15) // result => 16

ceiling(arr, 10) // result => 14

- if target is found it simply return $\overset{\text{target}}{\text{mid}}$.
but,
- if not found then it returns a smallest number greater or equal to target.

Let's check

$[\underset{s}{\overset{0}{2}}, \overset{1}{3}, \overset{2}{5}, \underset{m}{\overset{3}{\boxed{9}}}, \overset{4}{14}, \overset{5}{16}, \underset{e}{\overset{6}{18}}]$ target = 15

mid = $\frac{s+e}{2} = \frac{0+6}{2} = 3$

15 > 9 => ↓ start = mid+1;

$[\underset{s}{\overset{4}{14}}, \underset{m}{\overset{5}{16}}, \underset{e}{\overset{6}{18}}]$:

15 < 16 => ↓ end = mid - 1;

$[\underset{s\ m\ e}{14}]$

15>14

↓ start = mid+1

loop break // target = not found

↓

at this time start becomes greater than end. and loop is violated and see below what happened.

```
 e   s ↓
 4↓  5↓   6
[14, 16, 18]       start > end => loop
                                  break
```

Now,

- see where start is pointing to it is pointing to element which is greater than target and also smallest element.

- so answer is $\boxed{16}$ at index $\boxed{5}$

<u>let's</u> <u>code</u>

```
*   ceiling( int[] arr, int target) {

        int start = 0;
        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{
        return mid; // if target found
                       return index.
    }
  }

  return start; // if target is not found
}                 than it returns smallest
                  number greater or equal to target.
```