* Questions on Binary Search algorithm:

I] Ceiling of Number:

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

```
          0  1  2  3  4  5  6
arr = [ 2, 3, 5, 9, 14, 16, 18 ]    target = 15
```

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

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

- if target is found it simply return mid. (target ↓)
but,
- if not found then it returns a smallest number greater or equal to target.

Let's check

```
          0  1  2  3  4  5  6
[ 2, 3, 5, 9, 14, 16, 18 ]    target = 15
  s->      m          <-e
```

mid = (s+e)/2 = (0+6)/2 = 3

15 > 9 => ↓ start = mid+1;

```
          4  5  6
[ 14, 16, 18 ]  =
  s->  m  <-e
```

15 < 16 => ↓ end = mid - 1;

```
[ 14 ]
 s  m  e
```

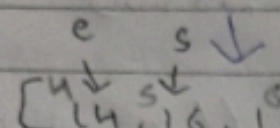
15 > 14

↓ 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.



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 16 at index 5

let's code

```
* 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.
```