3] *

Smallest ~~Letter~~ Letter Greater than targ

- Smallest letter greater than target

- Exact same approach like ceiling of number

- but ignore the equal to part (target == what we are looking (mid)) X

- arr = ['c', 'd', 'g', 'j']
  0 1 2 3

  target = 'j'

  smallestLetter(arr, 'c') => // 'd' ← ans

  smallestLetter(arr, 'g') => // 'j' ← ans

  smallestLetter(arr, 'j') => // 'c' ← ans
  ↑

- if letter not found, just return first letter

↓

formula = start % length for returning first letter

↓

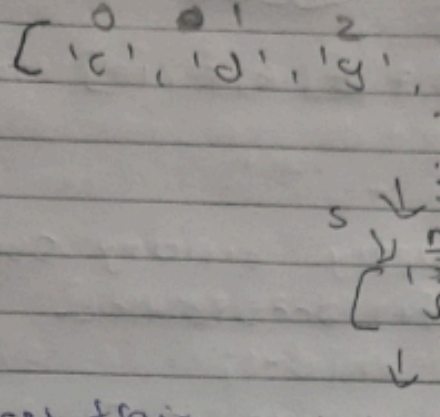
How let's see:

when condition is violated or break:

↓

~~s~~ start > end

- if target is a least letter



at this point = 3 + 1 = 4

start becomes length of arr

s > e => start == length => true

apply formula!!

start % length

4 % 4 = 0

so, it return 0 index which is first letter

let's code

```
smallestLetter(char[] arr, char 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{
            end = mid-1;
        }
    }
    return arr[start % arr.length];
}
```