* **BS LOWER BOUND** $\qquad$ UB

↳ <

The smallest index such that the number at that index is $>=$ given number.

eg:

$$arr[] = [\overset{0}{3}, \overset{1}{5}, \overset{2}{8}, \overset{3}{15}, \overset{4}{19}] \qquad n = 5$$

$$x = 8$$

→ ans:

3	5	8	15	19

↑ ↑ ↑
× ×

$8 \ge 8$

$8 >= 8$

∴ index = 2

→ $x = 9$

$\overset{0}{3}$	$\overset{1}{5}$	$\overset{2}{8}$	$\overset{3}{15}$	$\overset{4}{19}$

↳ ↑ ↑ ↑ ↑
× ×
$3 \ge 9$ $8 \ge 9$

$15 >= 9$

∴ index = 3

→ $x = 16$

$\overset{0}{3}$	$\overset{1}{5}$	$\overset{2}{8}$	$\overset{3}{15}$	$\overset{4}{19}$

× × × × ✓

$19 >= 16$ ✓

index = 4

→ $x = 20$

3	5	8	15	19

× × × × ×

$lb = 5$

= size of arr

= hypothetical case

example 2:

$$arr[] = \begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ [\,1 & 2 & 3 & 3 & 4\,7 & 8 & 9 & 9 & 9 & 11\,] \end{array}$$

↑ low   ↑ mid   ↑ high

(x=1)

ans = 10   // length of arr

mid = 0 + 9/2 = 7 ≥ 1

→ I did not get my lower bound, but I
at least got a number which is ≥ 1. This
can be my answer.

→ Now we found the mid, we need to go
left because we need to find LB.
∴ Eliminate the right search place & high = mid - 1
                                      ans = 4

$$[] = \begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 \\ [\,1 & 2 & 3 & 3 & 7 & 8 & 9 & 9 & 9 & 11\,] \end{array}$$

↑ ↑   ↑
low mid   high

mid = 0 + 3/2 = 1th 2 ≥ 1 ✓   ans = 1

this might be the answer but then to
search left coz we want lowest bound.
now high = mid - 1

$$[] = \begin{array}{cccc} 0 & 1 & 2 & 3 \\ [\,1 & 2 & 3 & 3 & 7 & \cdots \cdots\,] \end{array}$$

↑
low

↑
high

mid = $\frac{0+0}{0}$ = 0th = 1   1 ≥ 1

Still look left : hig = mid - 1 = -1 → high crosses low
                                              STOP

*

Floor and Ceil

↓	↓

largest no in   Smallest no in
array $<= x$ array $>= x$

eg: arr = [10, 20, 30, 40, 50]
$x = 25$
           ↑   ↑
         floor   ceil

eg: arr = [10, 20, 25, 30, 40]
$x = 25$
               ↑
            ceil
            ↑
         UB(upper bound)

→ code

```
floor ( arr, n) {
low= 0, high= n-1, ans = -1;
while ( low <= high) {
mid = (low+high) /2;
if ( a [mid ] <= mid )
    ans= arr [mid ]
    low = mid +1;
else
    high = mid -1
}
```

{ LB }

* LB and UB

int arr = [4, 6, 10, 12, 18, 20]

int x = 11   ans = 10

LB:

	0	1	2	3	4	5
	4	6	10	12	18	20

↑ L   ↑ M   ↑ H

x <= M ? x

L = M+1

	0	1	2	3	4	5
	4	6	10	12	18	20

↑ L   ↑ M   ↑ H

x <= mid

return high

x <= M : 11 <= 18 ✓ → H = M - 1

	0	1	2	3	4	5
	4	6	10	12	18	20

↑ L,H,M   M = 3+3/2 = 3

x <= M ? 11 <= 12 ? ✓ → high = mid - 1

	0	1	2	3	4	5
	4	6	10	12	18	20

↑ H   ↑ L,M

L = 3

H = 2

L <= H : 3 <= 2 ✗ → end loop.