QUESTION NO 1

SMALLEST LETTER GREATER THAN TARGET

You are given an array of characters `letters` that is sorted in **non-decreasing order**, and a character `target`. There are **at least two different** characters in `letters`.

Your function should return the smallest character in `letters` that is lexicographically greater than target. If such a character does not exist, return the first character in `letters`.

∴ You must achieve the solution with O(log(n))

Example 1:

```
Input: letters = ["c","f","j"], target = "a"

Output: "c"

Explanation: The smallest character that is lexicographically greater than 'a' in letters is 'c'.
```

Example 2:

```
Input: letters = ["c","f","j"], target = "c"

Output: "f"

Explanation: The smallest character that is lexicographically greater than 'c' in letters is 'f'.
```

Example 3:

```
Input: letters = ["x","x","y","y"], target = "z"

Output: "x"

Explanation: There are no characters in letters that is lexicographically greater than 'z' so we

return letters[0].
```

Constraints:

- `2 <= letters.length <= 10`^{4}
- `letters[i]` is a lowercase English letter
- `letters` is sorted in **non-decreasing** order
- `letters` contains at least two different characters
- `target` is a lowercase English letter

QUESTION NO 2

CLIMBING STAIRS

You are climbing a staircase. It takes `n` steps to reach the top. Each time you can either climb `1` or `2` steps. In how many distinct ways can you climb to the top?

Example 1:

```
Input: n = 2

Output: 2

Explanation: There are two ways to climb to the top.

1. 1 step + 1 step

2. 2 steps
```

Example 2:

```
Input: n = 3

Output: 3

Explanation: There are three ways to climb to the top.

1. 1 step + 1 step + 1 step

2. 1 step + 2 steps

3. 2 steps + 1 steps
```

Constraints:

- `1 <= n <= 45`

HINT

Fibonacci Series: The Fibonacci sequence is a type series where each number is the sum of the two that precede it. It starts from 0 and 1 usually. The Fibonacci sequence is given by 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, and so on.