Problem 1894: Find the Student that Will Replace the Chalk

Problem Information

Difficulty: Medium
Acceptance Rate: 53.17%
Paid Only: No
Tags: Array, Binary Search, Simulation, Prefix Sum

Problem Description

There are `n` students in a class numbered from `0` to `n - 1`. The teacher will give each student a problem starting with the student number `0`, then the student number `1`, and so on until the teacher reaches the student number `n - 1`. After that, the teacher will restart the process, starting with the student number `0` again.

You are given a **0-indexed** integer array `chalk` and an integer `k`. There are initially `k` pieces of chalk. When the student number `i` is given a problem to solve, they will use `chalk[i]` pieces of chalk to solve that problem. However, if the current number of chalk pieces is **strictly less** than `chalk[i]`, then the student number `i` will be asked to **replace** the chalk.

Return _the**index** of the student that will **replace** the chalk pieces_.

Example 1:

Input: chalk = [5,1,5], k = 22 **Output:** 0 **Explanation:** The students go in turns as follows: - Student number 0 uses 5 chalk, so k = 17. - Student number 1 uses 1 chalk, so k = 16. - Student number 2 uses 5 chalk, so k = 11. - Student number 0 uses 5 chalk, so k = 6. - Student number 1 uses 1 chalk, so k = 5. - Student number 2 uses 5 chalk, so k = 0. Student number 0 does not have enough chalk, so they will have to replace it.

Example 2:

Input: chalk = [3,4,1,2], k = 25 **Output:** 1 **Explanation:** The students go in turns as follows: - Student number 0 uses 3 chalk so k = 22. - Student number 1 uses 4 chalk so k =

18. - Student number 2 uses 1 chalk so k = 17. - Student number 3 uses 2 chalk so k = 15. - Student number 0 uses 3 chalk so k = 12. - Student number 1 uses 4 chalk so k = 8. - Student number 2 uses 1 chalk so k = 7. - Student number 3 uses 2 chalk so k = 5. - Student number 0 uses 3 chalk so k = 2. Student number 1 does not have enough chalk, so they will have to replace it.

Constraints:

* `chalk.length == n` * `1 <= n <= 105` * `1 <= chalk[i] <= 105` * `1 <= k <= 109`

Code Snippets

C++:

```
class Solution {
public:
int chalkReplacer(vector<int>& chalk, int k) {

}
};
```

Java:

```
class Solution {
public int chalkReplacer(int[] chalk, int k) {

}
}
```

Python3:

```
class Solution:
def chalkReplacer(self, chalk: List[int], k: int) -> int:
```