1894. Find the Student that Will Replace the Chalk

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 <= 10`^{5}
- `1 <= chalk[i] <= 10`^{5}
- `1 <= k <= 10`^{9}

1894. Find the Student that Will Replace the Chalk

```
/*
    Simulation
    Time complexity: O(kn) - TLE
    Space complexity: O(1)
*/
typedef std::vector<int> vi;
class Solution {
    public:
        int chalkReplacer(vi& chalk, int k) {
            int n=chalk.size();

            int ans=0;
            while(k>=0){
                for(int i=0;k>=0 && i<n;++i){
                    ans=i;
                    k-=chalk[i];
                }
            }

            return ans;
        }
};
```

1894. Find the Student that Will Replace the Chalk

```
/*
    Math
    Time complexity: O(n+n)=O(n) - AC
    Space complexity: O(1)
*/
typedef std::vector<int> vi;

class Solution {

    public:

        int chalkReplacer(vi& chalk, int k){

            int n=chalk.size();

            long long s=std::accumulate(chalk.begin(),chalk.end(),0*1ll);

            long long rem=(k*1ll)%s;

            k=(int)rem;

            for(int i=0;i<n;++i){

                k-=chalk[i];

                if(k<0) return i;

            }

            return -1; // Never reached

        }

};
```