Problem 2217: Find Palindrome With Fixed Length

Problem Information

Difficulty: Medium
Acceptance Rate: 37.70%
Paid Only: No
Tags: Array, Math

Problem Description

Given an integer array `queries` and a **positive** integer `intLength`, return _an array_ `answer` _where_ `answer[i]` _is either the_ `queries[i]th` _smallest**positive palindrome** of length_ `intLength` _or_ `-1` _if no such palindrome exists_.

A **palindrome** is a number that reads the same backwards and forwards. Palindromes cannot have leading zeros.

Example 1:

Input: queries = [1,2,3,4,5,90], intLength = 3 **Output:** [101,111,121,131,141,999] **Explanation:** The first few palindromes of length 3 are: 101, 111, 121, 131, 141, 151, 161, 171, 181, 191, 202, ... The 90th palindrome of length 3 is 999.

Example 2:

Input: queries = [2,4,6], intLength = 4 **Output:** [1111,1331,1551] **Explanation:** The first six palindromes of length 4 are: 1001, 1111, 1221, 1331, 1441, and 1551.

Constraints:

* `1 <= queries.length <= 5 * 104` * `1 <= queries[i] <= 109` * `1 <= intLength <= 15`

Code Snippets

C++:

```
class Solution {
public:
vector<long long> kthPalindrome(vector<int>& queries, int intLength) {

}
};
```

Java:

```
class Solution {
public long[] kthPalindrome(int[] queries, int intLength) {

}
}
```

Python3:

```
class Solution:
def kthPalindrome(self, queries: List[int], intLength: int) -> List[int]:
```