Coding Test: Number Padding Consistency Checker

Objective:

You are required to implement a TypeScript function that checks for consistent number padding in a sequence of strings. The purpose is to determine if the numbers are consistently left-padded with "0" characters to a specified length, if the numbers are unpadded, or if there is padding but it is inconsistent.

Background:

In certain contexts, such as Bates numbering, numerical IDs are often left-padded with zeros to a fixed length so that all numbers in a sequence maintain uniform length. Sometimes, these numbers might overflow beyond their padding, leading to variations in string length, even if the padding is infact applied consistently. This function should analyze the provided strings to detect consistent padding and return the observed padding length.

Task:

Complete the implementation of the `checkNumberPadding` function. This function should:

- **Analyze the padding consistency**: Check if the given strings are consistently padded with zeros.
- **Return a positive number (>1)**: If a padding length can be determined and is consistent. For example:
  - `["001", "002"]` should return `3`
  - `["001", "002", "9999"]` should return `3` (note that although the last string has length 4, it is still a valid & consistent input for a left-pad of length 3).
- **Return 1**: If it can be known that no padding was used (effectively the same as a padding length of 1). For example:
  - `["1", "2", “999”]` should return `1`
- **Return a negative number (<-1)**: If padding is not observed, but there isn’t enough information to say conclusively that it wasn’t just a small value. The absolute value of this number should indicate the smallest observed non-padded length. For example:
  - `["999", "9999"]` should return `-3` (padding length may be 3 or 2 or 1 (no padding), but it's inconclusive with the given information)
  - `["99", "999", "9999"]` should return `-2` (padding length may be 2 or 1 (no padding), but it's inconclusive with the given information)
- **Special-case return values**: Note that the above definitions leave room for a return value of `-1` and `0` without a defined meaning. Use `-1` to indicate inconsistent padding, and use `0` to indicate no observations.
  - `["01", "002"]` should return `-1` (inconsistent padding length)
  - `[]` should return `0` (nothing to observe)

Code Snippet:

typescript
Copy code

```
export function checkNumberPadding(intStrs: Iterable<string>): number
{
    // Implementation here
}
```

Submission Requirements:

1. **Implementation**: The function must be implemented in TypeScript.
2. **Submission**: Submit your completed code via a public GitHub repository.
3. **Documentation**: Include a README file with clear instructions on how to run your code.
4. **Testing**: Provide a testing script or detailed instructions on how to test your implementation.
5. **Integrity**: This is an individual test—external assistance is not allowed.