1317. Convert Integer to the Sum of Two No-Zero Integers

No-Zero integer is a positive integer that does not contain any 0 in its decimal representation.

Given an integer n, return a list of two integers [a, b] where:

a and b are No-Zero integers. a + b = n The test cases are generated so that there is at least one valid solution. If there are many valid solutions, you can return any of them.

Example 1:

Input: n = 2 Output: [1,1] Explanation: Let a = 1 and b = 1. Both a and b are no-zero integers, and a + b = 2 = n. Example 2:

Input: n = 11 Output: [2,9] Explanation: Let a = 2 and b = 9. Both a and b are no-zero integers, and a + b = 11 = n. Note that there are other valid answers as [8, 3] that can be accepted.

Constraints:

2 <= n <= 104

First approach

```
/**
 * @param {number} n
 * @return {number[]}
 */
var getNoZeroIntegers = function(n) {

    function hasZero(num) {
        return num.toString().includes("0");
    }

    for (let i = 1; i < n; i++) {
            let j = n - i;
            if (!hasZero(i) && !hasZero(j)) {
                return [i, j];
            }
        }
        return 0;
};
```

Second approach

```
var getNoZeroIntegers = function(n) {
    function hasZero(num) {
        while (num > 0) {
            if (num % 10 === 0) return true;
            num = Math.floor(num / 10);
        }
        return false;
    }

    for (let i = 1; i < n; i++) {
        let j = n - i;
        if (!hasZero(i) && !hasZero(j)) {
            return [i, j];
        }
    }
};
```

Third approach

```
var getNoZeroIntegers = function(n) {
    function hasZero(num) {
        while (num > 0) {
            if (num % 10 === 0) return true;
            num = Math.floor(num / 10);
        }
        return false;
    }


    let a = 1;
    let b = n - a;

    while (hasZero(a) || hasZero(b)) {
        a++;
        b = n - a;
    }

    return [a, b];
};
```