Leetcode Problem 1. (Easy)

Plus One

You are given a **large integer** represented as an integer array digits, where each digits[i] is the i^{th} digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.

Increment the large integer by one and return *the resulting array of digits*.

Example 1:

Input: digits = [1,2,3]
Output: [1,2,4]
Explanation: The array represents the integer 123.
Incrementing by one gives 123 + 1 = 124.
Thus, the result should be [1,2,4].

Example 2:

Input: digits = [4,3,2,1]
Output: [4,3,2,2]
Explanation: The array represents the integer 4321.
Incrementing by one gives 4321 + 1 = 4322.
Thus, the result should be [4,3,2,2].

Example 3:

Input: digits = [9]
Output: [1,0]
Explanation: The array represents the integer 9.
Incrementing by one gives 9 + 1 = 10.
Thus, the result should be [1,0].

Constraints:

- 1 <= digits.length <= 100
- 0 <= digits[i] <= 9
- digits does not contain any leading 0's.

https://leetcode.com/problems/plus-one/

```
class Solution {
    public int[] plusOne(int[] digits) {
    int carry = 1;
```

```
        int n = digits.length;
                for (int i = n - 1; i >= 0; i--) {
            int sum = digits[i] + carry;
            digits[i] = sum % 10;
            carry = sum / 10;

                        if (carry == 0) {
                return digits;
            }
        }

        int[] result = new int[n + 1];
        result[0] = carry;
        for (int i = 1; i <= n; i++) {
            result[i] = digits[i - 1];
        }

        return result;
    }
}
```