

You are given a large integer represented as an integer array `digits`, where each `digits[i]` is the  $i^{\text{th}}$  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:

- $1 \leq \text{digits.length} \leq 100$
- $0 \leq \text{digits}[i] \leq 9$
- `digits` does not contain any leading 0's.

Approach :

Step - 1 : count from LSB to MSB.

Step - 2 : if LSB is equal to 9 then it returns 0 and set LSB to 1 because of carry..  
If LSB is less than 9 then it increment MSB by 1.