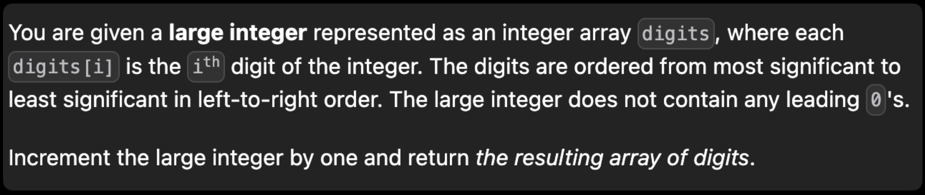
Leetcode Logic building Day 1 / 60

Leetcode 66. PLUS ONE

Amazon, Apple, Adobe, Google



You are given a **large integer** represented as an integer array `digits`, where each `digits[i]` is the `ith` 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*.

for "1 2 3"

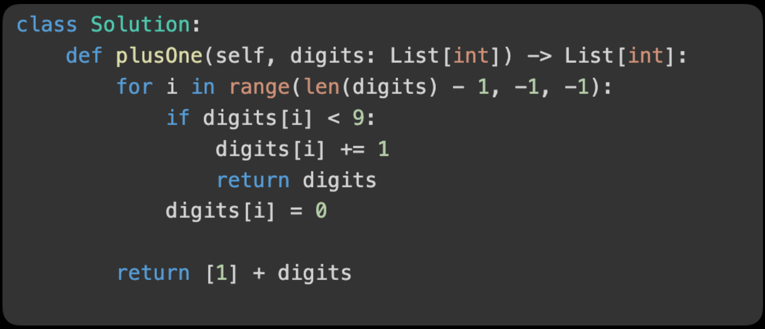
Input	1	2	3
Output	1	2	4

Approach -1

1) If the digits[i]<9 just add 1 and return

2) If it's equal to 9 make it zero & let the loop run

3) If loop runs and everything is 0 just add [1] in the front

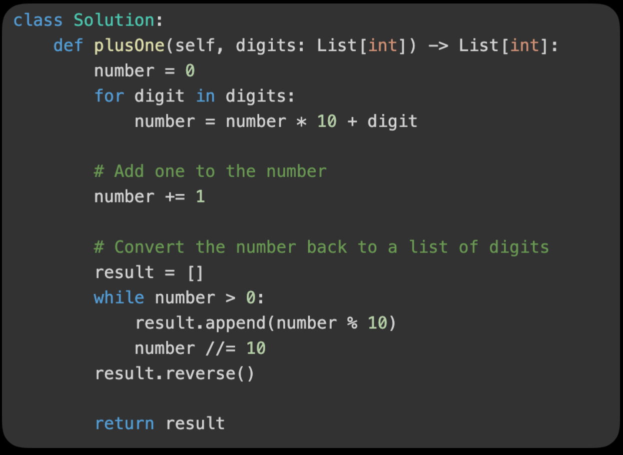


```
class Solution:
    def plusOne(self, digits: List[int]) -> List[int]:
        for i in range(len(digits) - 1, -1, -1):
            if digits[i] < 9:
                digits[i] += 1
                return digits
            digits[i] = 0

        return [1] + digits
```

Approach 2 - Not optimal but intutive

- We change the array to number
- Then we add 1
- then break the number in digits and convert the number back to the list
- and now reverse the list



```
class Solution:
    def plusOne(self, digits: List[int]) -> List[int]:
        number = 0
        for digit in digits:
            number = number * 10 + digit

        # Add one to the number
        number += 1

        # Convert the number back to a list of digits
        result = []
        while number > 0:
            result.append(number % 10)
            number //= 10
        result.reverse()

        return result
```