## Add Digits

Given a non-negative integer `num`, repeatedly add all its digits until the result has only one digit.

For example:

Given `num = 38`, the process is like: `3 + 8 = 11`, `1 + 1 = 2`. Since `2` has only one digit, return it.

**Follow up:**
Could you do it without any loop/recursion in O(1) runtime?

1. A naive implementation of the above process is trivial. Could you come up with other methods?
2. What are all the possible results?
3. How do they occur, periodically or randomly?
4. You may find this Wikipedia article useful.

**Credits:**
Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

## Solution 1

The problem, widely known as *digit root* problem, has a congruence formula:

```
https://en.wikipedia.org/wiki/Digital_root#Congruence_formula
```

For base $b$ (decimal case $b = 10$), the digit root of an integer is:

- dr($n$) = 0 if $n$ == 0
- dr($n$) = ($b$-1) if $n$ != 0 and $n$ % ($b$-1) == 0
- dr($n$) = $n$ mod ($b$-1) if $n$ % ($b$-1) != 0

or

- dr($n$) = 1 + ($n$ - 1) % 9

Note here, when $n$ = 0, since ($n$ - 1) % 9 = -1, the return value is zero (correct).

From the formula, we can find that the result of this problem is immanently periodic, with period ($b$-1).

Output sequence for decimals ($b$ = 10):

~input: 0 1 2 3 4 ...
output: 0 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 ....

Henceforth, we can write the following code, whose time and space complexities are both $O$(1).

```
class Solution {
public:
    int addDigits(int num) {
        return 1 + (num - 1) % 9;
    }
};
```

Thanks for reading. :)

written by zhiqing_xiao original link here

## Solution 2

```
int addDigits(int num) {
    int res = num % 9;
    return (res != 0 || num == 0) ? res : 9;
}
```

The essence of this problem is that 10^n ≡ 1 (mod 9), and thus a*n*10^*n* + ... + *a*1*10 + a_0 ≡ a_n + ... + a_1 + a_0 (mod 9). This process can be continued until a number less than 9 is gotten, i.e. num % 9. For any digit n, n = n % 9 unless n = 9. The only confusing case is n % 9 = 0, but addDigits(num) = 0 if and only if num = 0, otherwise it should be 9 in fact.

written by DeKoder original link here

## Solution 3

If an integer is like 100a+10b+c, then (100a+10b+c)%9=(a+99a+b+9b+c)%9=(a+b+c)%9

```
class Solution:
    # @param {integer} num
    # @return {integer}
    def addDigits(self, num):
        if num==0:
            return 0
        return num%9 if num%9!=0 else 9
```

written by yawn.zheng original link here