Note: please follow the framework

1. Understand the question
2. Understand Input / Output
3. Prepare the solution in rough
4. Write clean code in book
5. Dry run and write output
6. Match the output
7. Write code in laptop and run, if output not matching debug and fix

Question 1:

Given a **positive** integer n, return *the smallest positive integer that is a multiple of **both** 2 and* n

Example 1:

Input: n = 5
Output: 10
Explanation: The smallest multiple of both 5 and 2 is 10
Example 2:

Input: n = 6
Output: 6
Explanation: The smallest multiple of both 6 and 2 is 6. Note that a number is a multiple of itself

Question 2:

You are given positive integers n and m.

Define two integers, num1 and num2, as follows:

- num1: The sum of all integers in the range [1, n] that are **not divisible** by m.
- num2: The sum of all integers in the range [1, n] that are **divisible** by m.

Return *the integer* num1 - num2.

Example 1:

Input: n = 10, m = 3
Output: 19
Explanation: In the given example:

- Integers in the range [1, 10] that are not divisible by 3 are [1,2,4,5,7,8,10], num1 is the sum of those integers = 37.
- Integers in the range [1, 10] that are divisible by 3 are [3,6,9], num2 is the sum of those integers = 18.
We return 37 - 18 = 19 as the answer.
Example 2:

Input: n = 5, m = 6
Output: 15
Explanation: In the given example:
- Integers in the range [1, 5] that are not divisible by 6 are [1,2,3,4,5], num1 is the sum of those integers = 15.
- Integers in the range [1, 5] that are divisible by 6 are [], num2 is the sum of those integers = 0.
We return 15 - 0 = 15 as the answer.

Question 3:

Given an integer number n, return the difference between the product of its digits and the sum of its digits.

Example 1:

Input: n = 234
Output: 15
Explanation:
Product of digits = 2 * 3 * 4 = 24
Sum of digits = 2 + 3 + 4 = 9
Result = 24 - 9 = 15

Example 2:

Input: n = 4421
Output: 21
Explanation:
Product of digits = 4 * 4 * 2 * 1 = 32
Sum of digits = 4 + 4 + 2 + 1 = 11
Result = 32 - 11 = 21

Question 4:

Given an integer num, return *the number of steps to reduce it to zero*.

In one step, if the current number is even, you have to divide it by 2, otherwise, you have to subtract 1 from it.

Example 1:

Input: num = 14
Output: 6
Explanation:
Step 1) 14 is even; divide by 2 and obtain 7.
Step 2) 7 is odd; subtract 1 and obtain 6.
Step 3) 6 is even; divide by 2 and obtain 3.
Step 4) 3 is odd; subtract 1 and obtain 2.
Step 5) 2 is even; divide by 2 and obtain 1.
Step 6) 1 is odd; subtract 1 and obtain 0.

Example 2:

Input: num = 8
Output: 4
Explanation:
Step 1) 8 is even; divide by 2 and obtain 4.
Step 2) 4 is even; divide by 2 and obtain 2.
Step 3) 2 is even; divide by 2 and obtain 1.
Step 4) 1 is odd; subtract 1 and obtain 0.

Question 5:

A **self-dividing number** is a number that is divisible by every digit it contains.

- For example, 128 is **a self-dividing number** because 128 % 1 == 0, 128 % 2 == 0, and 128 % 8 == 0.

A **self-dividing number** is not allowed to contain the digit zero.

Given two integers left and right, return *a list of all the* ***self-dividing numbers*** *in the range* [left, right].

Example 1:

Input: left = 1, right = 22

Output: [1,2,3,4,5,6,7,8,9,11,12,15,22]

Example 2:

Input: left = 47, right = 85
Output: [48,55,66,77]