1. GFG: Check Status

This sounds like a prompt that might refer to checking whether a number is even or odd—maybe you meant “Check Status” to imply the status of a number? If so, here’s how it goes:

- **Basic logic**: Use the modulus operator: if `n % 2 == 0`, it's even; otherwise, it's odd.
- **Efficient trick**: Use bitwise AND: `(n & 1) == 0` means even, since the last binary bit of even numbers is 0.GeeksforGeeksStack Overflow

---

2. GFG: Mark Even and Odd (Using Conditional Statements)

Here, we use conditionals—`if…else` or shorthand constructs—to check numbers:

Using `if…else`:

```
if n % 2 == 0:
    print("Even")
else:
    print("Odd")
```

- CodefinityVultr Docs

Using ternary (short-hand):

```
result = "Even" if n % 2 == 0 else "Odd"
print(result)
```

- GeeksforGeeksVultr Docs

Think of it this way: the conditional is the narrative fork—a single line that reveals which path you take, even or odd.

---

3. GFG: If Conditional Statement

This likely refers to the **theory or explanation** of `if` statements (possibly in Python, as per the GFG tutorial):

- The `if` statement executes a block of code if a condition is true.
- Variants include:
    - **`if … else`** for two branches
    - **`elif`** for multiple conditions
    - **Nested `if…else`**
    - **Shorthand single-line `if`** (e.g., `print(...) if condition else ...`)
    - **Match–case** (Python's switch-like structure)GeeksforGeeks

Each is a stylistic choice in storytelling—setting the scene, hanging on tension, resolving the arc.

---

4. LeetCode Q1: 258. Add Digits

Now, the LeetCode hero emerges: **Add Digits**, where you repeatedly sum a number's digits until a single digit remains.

Two Approaches to Solve It:

A. Iterative (Loop) Method:
Keep summing digits until you end up with a single-digit number:

```
while num > 9:
    s = 0
    while num:
        s += num % 10
        num //= 10
    num = s
return num
```

- tenderleo.gitbooks.ioDesign Gurus

This is like the traditional climb—step by step.

B. Mathematical Shortcut — Digital Root:
Use the formula:

```
result = 0 if num == 0 else (num - 1) % 9 + 1
```

- This uses the magic of modulo 9, yielding an O(1), constant-time solution.AlgoMonsterDesign GurusWalkCCC

Think of it as the secret portal that bypasses the climb—straight to the summit.

Here's a Storytelling-Infused Summary:

- You begin with a number—our main character.
- You ask: "Even or odd?" It checks itself via modulus or bitwise (simple and direct).
- You use an **`if` statement** to reveal the label—there's your conflict and resolution.
- In **Add Digits**, instead of walking each step, you use the digital root—cutting through complexity and arriving elegantly at the final digit.

Code Illustrations

Even/Odd Check (Python)

```
def status(n):
    # Conditional check with if…else
    return "Even" if n % 2 == 0 else "Odd"
```

Add Digits (Python)

```
def add_digits(num):
    # Optimal mathematical approach
    return 0 if num == 0 else (num - 1) % 9 + 1
```