**Bit Manipulation**

**✅ 1. Theory Questions (Basic ➝ Advanced)**

**🔹 Basic**

1. What is bit manipulation?
2. How are integers stored in memory in Java?
3. What is the difference between **&**, **|**, **^**, **~**, **<<**, and **>>** operators?
4. What’s the use of bitwise AND/OR for masking?
5. How does left shift (<<) multiply and right shift (>>) divide?

**🔹 Intermediate**

1. How do you check if a number is even or odd using bitwise operators?
2. What is the difference between signed and unsigned right shift (>> vs >>>) in Java?
3. What is a bitmask? How is it used?
4. How do you count the number of 1's in a binary number?

**🔹 Advanced**

1. What is the concept of a power of two using bit manipulation?
2. How can you swap two numbers using XOR?
3. What is the XOR property and its use in problems like "single number"?
4. How can you toggle, set, or clear a specific bit?
5. What are common bit tricks used in competitive programming?

**💻 2. Coding Questions (Basic ➝ Advanced)**

**🔹 Basic**

1. Check if a number is even or odd
2. Get the i-th bit of a number
3. Set the i-th bit of a number
4. Clear the i-th bit of a number
5. Toggle the i-th bit of a number

**🔹 Intermediate**

1. Count the number of set bits (1’s) in a number
2. Check if a number is a power of 2
3. Fast exponentiation using bit manipulation
4. Clear last i bits
5. Clear a range of bits (i to j)

**🔹 Advanced**

1. Find the only non-repeating element in an array where every other appears twice (using XOR)
2. Find 2 non-repeating elements in an array where every other appears twice
3. Find the element that appears once where every other appears three times
4. **Generate all subsets of a set using bits**
5. Add two integers without using + or - operator
6. Find XOR of numbers from 1 to n
7. Find the position of the rightmost set bit
8. Count total bits required to convert A to B

**🧠 Extra Tips to Master Bit Manipulation:**

* Understand **XOR** deeply — it's used in many clever tricks.
* Practice binary representation of numbers (convert decimal ↔ binary).
* Memorize how masks work to **set, clear, toggle bits**.
* Know how shifting left/right affects the value.