☑️ Miscellaneous DSA Concepts: Bitwise, Scope, and Modifiers

🫠 Bitwise Operators

Operate on binary representations of integers.

- **AND (&)**: 1 if both bits are 1
  `5 & 3 → 101 & 011 = 001 → 1`

- **OR (|)**: 1 if either bit is 1
  `5 | 3 → 101 | 011 = 111 → 7`

- **XOR (^)**: 1 if bits are different
  `5 ^ 3 → 101 ^ 011 = 110 → 6`

🔁 Shift Operators

Move bits left/right → used for multiplication/division by powers of 2.

- **Left Shift (<<)**: $x \ll n = x \times 2^n$
- **Right Shift (>>)**: $x \gg n = x \div 2^n$

🔢 Operator Precedence & Associativity

- **Precedence (high to low)**:
  - Unary operators (`++, --`)
  - Arithmetic (`*, /, %`)
  - Relational (`<, >, ==`)
  - Bitwise (`&, |, ^`)
  - Logical (`&&, ||`)
  - Assignment (`=`)
- **Associativity**:
  - Left to right for most binary operators
  - Use `()` to override precedence

🧪 Scope of Variables

- **Local Scope**: Declared inside functions/blocks → accessible only there.
- **Global Scope**: Declared outside all functions → accessible everywhere.

🧬 Data Type Modifiers

- `long` → increases data size (≥ 4 bytes)

- **short** → decreases size (used for small values like age)
- **unsigned** → only positive values, increases positive range
- **signed** → can store both positive and negative values

📑 Homework Tasks

1. Perform **AND, OR, XOR** operations on sample numbers manually and verify in code.
2. Check if a number is a **power of 2** (without loop).
3. **Reverse** a number and store it.