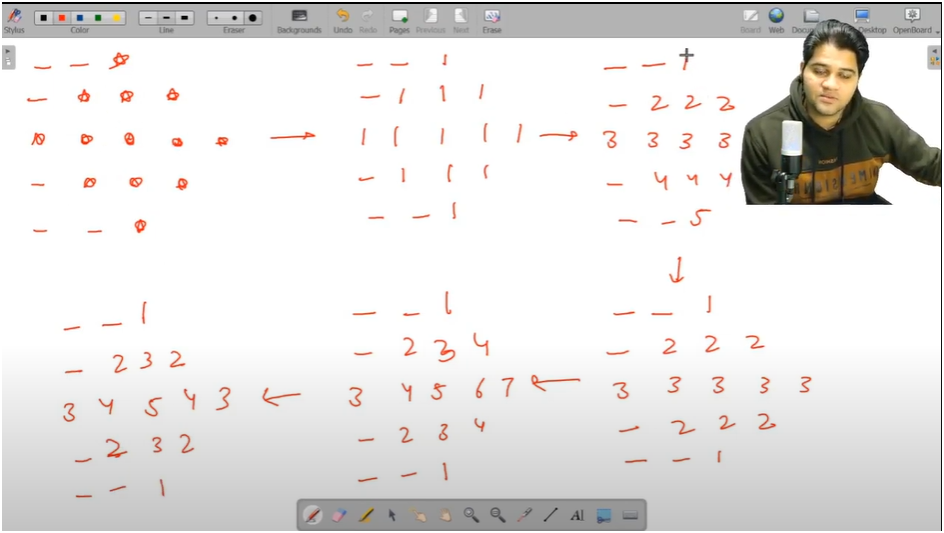


HOW TO SOLVE PROBLEM FLOW OF IT



**UNDERSTAND THE FLOW OF PATTERNS**

* **Start with Star Pattern**
  + **Begin with a simple \* (star) pattern layout.**
  + **Focus on the position/structure before adding any values.**
* **THEN REPLACE STARS WITH 1’S**

**Once the shape/pattern is confirmed with stars, replace each \* with 1.**

**This acts as a placeholder and helps understand the fixed shape logic.**

* **THEN REPALCE 1’S WITH CURRENT ROW NUMBERS**
  + **Convert every 1 in a row to the current row number.**
  + **🔁 For example:  
    If you’re on row i = 2, print 2 2 2 ... in that row.**
  + **This helps build uniform row value patterns.**
* **Before & After Middle Row Logic**
  + **Until the middle row, print values as the current row number.**
  + **After the middle row, start decreasing the row number back to 1.**
  + **✅ Example:  
    If total rows = 5, middle = 3**
    - **Row 1: 1**
    - **Row 2: 2**
    - **Row 3: 3**
    - **Row 4: 2 (decrement from 3)**
    - **Row 5: 1 (decrement again)**
* **Incremental Row Values (Left to Right within Row)**
  + **In more advanced patterns:**
  + **Start each row with a number k, then increment as you move right.**
  + **Example: 3 4 5 6 7  
    Here, k starts at 3 and increments with each print in the row.**
* **Final Pattern Logic (Symmetry in a Row)** **Final Pattern Logic (Symmetry in a Row)**
  + **NOW IF WE OBSERVED CAREFULLY IN LAST PATTERN TILL MIDDLE OF THE CURRENT ROW NUMBER IT IS INCREMENT BY1 THEN AFTER MIDDLE OF THE CURRENT ROW NUMBER IS IS DECREMENTING BY 1**
  + **MEANING**
  + **First half of each row increases (1 → mid)**
  + **Second half of each row mirrors the first in decreasing order.**
  + **Use an incrementing variable till mid of row**
  + **Then decrement it while printing the remaining values.**

**ALGORITHM:-**

**📌 PATTERN: NUMBER DIAMOND (CENTERED)**

**\* 🔢 INPUT:**

**\* - A single odd integer `total\_rows` (example: 5 or 7)**

**🧠 LOGIC BREAKDOWN:**

**\* 1. Initialize:**

**\* - `spaces = total\_rows / 2`: Number of spaces before numbers starts (for centering).**

**\* - `numbers = 1`: Total numbers to print in current row.**

**\* - `val = 1`: Starting number of the row.**

**\***

**\* 2. Outer Loop → runs from 1 to `total\_rows`:**

**\* - Handles each row of the pattern.**

**\* 3. Inner Loops:**

**\* 🔹 First Inner Loop (for spaces):**

**\* - Prints tabs `\t` to align numbers in center.**

**\* 🔹 Second Inner Loop (for numbers):**

**\* - Start from `val`, increment till the middle of the row,**

**\* then decrement after the middle.**

**\* - This creates the symmetrical number effect.**

**\* 4. Update Logic (middle check):**

**\* - First Half (i ≤ total\_rows / 2):**

**\* - Decrease spaces by 1**

**\* - Increase number count by 2**

**\* - Increase starting value (`val`) by 1**

**\* - Second Half:**

**\* - Reverse the above logic to shrink the diamond**

**\***

**✅ SAMPLE OUTPUT (for total\_rows = 5):**

**1**

**2 3 2**

**3 4 5 4 3**

**2 3 2**

**1**

**📌 KEY CONCEPTS:**

**\* - Symmetry: Built using mirror logic (increasing till mid, then decreasing)**

**\* - Centering: Achieved by adjusting spaces before printing numbers**

**\* - Row-wise dynamic value generation using `val` and `t`**