Event 5 - Reverse Coding Challenge Rulebook

Reverse Coding Challenge	
Location	Lab with Internet Access
2 Rounds	Qualifier and Finale
Per team	11 minutes
Max teams (team of 2): 15	2 hours
Prize	Point based (in the case of ties, time based)
Jury	External Expert

Event Overview: A Reverse Coding Challenge involves participants working backward from a given output or result to deduce the original code or algorithm that produced it. This type of challenge tests problem-solving skills, algorithmic thinking, and the ability to understand logic from the results rather than the process.

Participants will be provided with the final output (e.g., a specific set of results or a data structure) of an algorithm or program, and their task is to reverse engineer the code that could have produced this output. It's an excellent way to test debugging, logic-building, and reasoning skills, where the coder needs to "decode" the process used to create the output.

Stages: Qualifiers and Finale (top 4 teams compete in the finale)

Challenge Structure:

1. **Provided Output:**
    - Participants will receive a set of outputs. These can include:
        - A list of numbers
        - A string pattern or sequence
        - Data visualizations (e.g., graphs or tables)
        - A series of log outputs from a program

2. **Goal:**
    - Write the algorithm or code snippet that would generate the exact output provided.
    - The code must match the output in terms of the format, values, and any specific structure mentioned.

3. **Example Challenges:**
    - **Output 1:** A sequence of numbers with a specific pattern (e.g., Fibonacci series, prime numbers).
    - **Output 2:** A string with encoded characters or text (e.g., Caesar cipher, Morse code).
    - **Output 3:** A table or grid of data (e.g., matrix operations, sorting algorithms).
    - **Output 4:** A simple game result or score-based output.

4. **Additional Constraints:**
    - The algorithm must be efficient in terms of time and space complexity.
    - Participants can choose any programming language, but they must complete the challenge in the given time.

Scoring:

- **Correctness (60%):** How accurately the code matches the expected output.
- **Efficiency (20%):** How well the code handles performance in terms of time and space complexity.
- **Creativity (10%):** For those who go beyond the basic solution and introduce more elegant or optimized methods.
- **Code Readability (10%):** Code should be clean, well-commented, and easy to understand.

Example Challenge:

Output:

4 8 15 16 23 42

Task: Reverse engineer the algorithm that produces the above output.

Prizes and Recognition:

- **Winner:** 5,000 INR + Certificate
- **Runner-Up:** 3000 INR + Certificate