Function Analysis for EPC & ISP with CFG Updates

Objective

1. Develop **executable test scripts** for assigned functions while ensuring:
   - **Edge-Pair Coverage (EPC)** – For control-flow-driven logic **(requires CFG).**
   - **Partition-based White-box Coverage (PWC)** – Based on Input Space Partitioning (ISP).

     Input Domain Modeling (IDM) must first be performed using either:
     - Interface-based approach
     - Functionality-based approach
     - Explicitly define characteristics and corresponding input partitions (blocks) before applying PWC.
2. **Control Flow Graph (CFG) Updates** – Review and refine existing CFGs for accuracy, ensuring all branches, loops, and decision points are correctly represented.

Group Assignment

Each group is assigned two functions. Your Group Details is mentioned in the sheet provided. Your tasks below must be done **for both** functions.

Group	Assigned Functions
1	`cli_add_history`, `cli_int_free_buildmode`
2	`cli_enable`, `crypto_hash_sha256_update`
3	`crypto_auth_hmacsha256_init`, `cli_set_privilege`
4	`SHA256_Pad`, `crypto_auth_hmacsha256_verify`
5	`cli_set_configmode`, `be32dec_vect`
6	`cli_register_command`, `crypto_hash_sha256`
7	`crypto_auth_hmacsha256_update`, `cli_set_idle_timeout`
8	`cli_register_command_core`, `be32enc_vect`
9	`cli_build_shortest`, `crypto_auth_hmacsha256`
10	`doFCS`, `crypto_hash_sha256_final`
11	`cli_unregister_tree`, `SHA256_Transform`

12	crypto_auth_hmacsha256_final,cli_set_hostname

Tasks

- **Function Classification**
    - For **each assigned function**, determine the most suitable testing approach:
        - **EPC** (Edge Pair Coverage)
        - **ISP** (Input Space Partitioning)
        - Both
    - Clearly **justify** your decision based on:
        - Control Flow: Presence of branches, loops, or nested conditions.
        - Input Variability: Parameter types (numeric, strings, structs), ranges, and edge cases.
- **Test Script Development**
    - Design **executable test cases** to satisfy:
        - Edge-Pair Coverage (EPC)
            - Cover all feasible edge-pair paths in the CFG.
            - Include test data validating each decision path.
        - Partition-based White-box Coverage (PWC/ISP)
            - Define input partitions (valid/invalid ranges, boundary values).
            - Generate test values for each partition (e.g., int: [MIN, 0, MAX]).
    - **You must include actual test data values** in your report that satisfy:
        - Each edge-pair path **in EPC**
        - Each input partition **in ISP**
- **CFG Validation & Updates**
    - Create CFG of your assigned functions (EPC).
    - Also Review your **previous CFG diagrams**.
    - For each **function**, check:
        - Are all branches, loops, and decisions clearly shown?
        - Is the graph connected and semantically accurate?
    - **Update your CFGs** where necessary and include them in your report.

Submission Guidelines

- **Report (PDF)**
    - Test Evidence: Screenshots of executed test scripts.
    - CFGs: graphs + Updated graphs with clear node/edge labeling.
    - Input Justification: Explanation of EPC paths and ISP partitions.
    - Correct Input Values According to EPC and PWC.
- **Code (ZIP)**
    - Executable test scripts for both assigned functions.