### **CAIR Milestone Execution Plan**

### **📌 Accelerated Execution Plan for McEliece QC-MDPC (C++) in 20-25 Days**

Since we have a **tight deadline (20-25 days)** for implementing **one algorithm**, we need a **highly optimized and parallel execution strategy** for your **5 developers**.

## **🚀 Strategy: Fast Execution with Parallel Development**

1. **Break the work into 5 core components.**
2. **Assign each developer to work in parallel.**
3. **Run daily standups and code reviews to track progress.**
4. **Prioritize test-driven development (TDD) for minimal debugging overhead.**
5. **Deliver in 4 phases: Setup, Development, Testing, Documentation.**

# **📅 4-Phase Execution Plan (20-25 Days)**

| **Phase** | **Timeline** | **Task** |
| --- | --- | --- |
| **Phase 1** – Setup & Design | Day 1-3 | Finalize architecture, set up environment, define interfaces |
| **Phase 2** – Core Development | Day 4-12 | Implement KeyGen, Encrypt, Decrypt in parallel |
| **Phase 3** – Testing & Optimization | Day 13-18 | Unit, integration, performance & security testing |
| **Phase 4** – Documentation & Delivery | Day 19-25 | Finalize documentation, stakeholder review, bug fixes |

# **✅ Task Division Among 5 Developers**

| **Developer** | **Assigned Task** | **Key Deliverables** |
| --- | --- | --- |
| **Dev 1 (Algorithm Expert)** | Research QC-MDPC parameters & define cryptographic model | Mathematical formulation, function prototypes |
| **Dev 2 (KeyGen Developer)** | Implement **KeyGen()**: Generate public/private keys | Key generation logic using Crypto++ |
| **Dev 3 (Encryption Developer)** | Implement **Encrypt()**: Encode message, add controlled errors | Encryption module & encoding logic |
| **Dev 4 (Decryption Developer)** | Implement **Decrypt()**: Recover plaintext using error correction | Decoding algorithm implementation |
| **Dev 5 (Testing & Security Expert)** | Develop unit, integration, security tests | Performance benchmarks, attack simulations |

# **📅 Phase 1: Setup & Design (Day 1-3)**

**🎯 Goal:** Prepare architecture, define interfaces & set up environment.

✅ **Tasks:**

* **Dev 1:** Research QC-MDPC logic & finalize parameters.
* **Dev 2-4:** Write function prototypes & pseudocode.
* **Dev 5:** Define test cases for KeyGen, Encrypt, Decrypt.
* **All Devs:**
  + **Set up development environment**: Install **Crypto++**, **CMake**, **CLion/VSCode**.
  + **Create Git Repository & Branches** (main, dev, feature/\*).

✅ **Deliverables:**✔️ Function prototypes for KeyGen(), Encrypt(), Decrypt().  
✔️ Pseudocode & design flowchart.  
✔️ Setup Git branches & CI/CD pipeline.

# **📅 Phase 2: Core Development (Day 4-12)**

**🎯 Goal:** Implement all modules in parallel.

✅ **Tasks:**

* **Day 4-6: Implement Key Modules**
  + **Dev 2:** Implement KeyGen()
    - Use **Crypto++** for **random key generation**.
    - Generate **parity-check matrices** for QC-MDPC.
  + **Dev 3:** Implement Encrypt()
    - Encode plaintext using **public key**.
    - Introduce **controlled errors** for security.
  + **Dev 4:** Implement Decrypt()
    - Recover plaintext using **error correction**.
* **Day 7-9: Integration & Code Reviews**
  + **Merge KeyGen(), Encrypt(), Decrypt()** into the dev branch.
  + **Dev 5:** Start writing test cases (unit + integration tests).
* **Day 10-12: Refactoring & Optimization**
  + Optimize **key storage & retrieval mechanisms**.
  + Reduce **encryption/decryption latency**.

✅ **Deliverables:**✔️ **Working KeyGen, Encrypt, Decrypt functions**.  
✔️ **Integrated codebase with compiled executable**.

# **📅 Phase 3: Testing & Optimization (Day 13-18)**

**🎯 Goal:** Ensure correctness, security, and performance.

✅ **Tasks:**

* **Day 13-14: Unit Testing**

**Dev 5:** Test each function individually:  
cpp  
CopyEdit  
TEST\_CASE("KeyGen generates valid keys", "[KeyGen]") {

auto [pubKey, privKey] = KeyGen();

REQUIRE(!pubKey.empty());

REQUIRE(!privKey.empty());

}

* **Day 15-16: Integration & Security Testing**

**Validate encryption-decryption workflow**:  
scss  
CopyEdit  
Input plaintext → Encrypt() → Decrypt() → Verify output

* + **Simulate attacks** (quantum-inspired brute force, error injection).
  + **Check compliance with cryptographic standards**.
* **Day 17-18: Performance Benchmarking**
  + Measure:
    - **Execution time of KeyGen, Encrypt, Decrypt.**
    - **Memory usage & efficiency.**

✅ **Deliverables:**✔️ **All unit & integration tests pass.**✔️ **Performance & security benchmarks documented.**

# **📅 Phase 4: Documentation & Final Delivery (Day 19-25)**

**🎯 Goal: Write documentation, finalize the project & present.**

**✅ Tasks:**

* **Day 19-20: Technical Documentation**
  + **Dev 1: Document mathematical formulation of QC-MDPC.**
  + **Dev 2-4: Document KeyGen, Encrypt, Decrypt API specs.**
  + **Dev 5: Write test report with performance analysis.**
* **Day 21-22: Final Code Review & Stakeholder Presentation**
  + **Live demo of encryption-decryption workflow.**
  + **Showcase security resilience tests.**
* **Day 23-25: Final Handoff & Bug Fixes**
  + **Deliver:**
    - **Source code repository.**
    - **Test cases & reports.**
    - **User & developer documentation.**

**✅ Deliverables:  
✔️ Complete technical & user documentation.  
✔️ Final working version of McEliece QC-MDPC (C++).  
✔️ Code ready for stakeholder presentation.**

**📌 Summary of Execution Timeline**

**Phase Task Days**

**Phase 1 Setup, Research, & Design 1-3**

**Phase 2 Core Development (Parallel) 4-12**

**Phase 3 Testing & Optimization 13-18**

**Phase 4 Documentation & Delivery 19-25**