**SOLID Principal Delivery Notes**

The console application demonstrates the implementation of a Missing Number Finder using SOLID design principles and multiple algorithmic approaches.

A part from the original requirement, there are a few enhancements has been made to express my understanding of the SOLID principle.

* **SOLID Compliance**: Full implementation of all 5 SOLID principles.
* **Multiple Solutions**: 4 different algorithmic approaches to same problem.
* **Enhanced Requirements**: Finds all missing numbers, not just one.
* **Multiple Algorithm Support**: 4 different algorithmic approaches.
* **Flexible Architecture:** Runtime algorithm selection capability.
* **Testable Design**: Unit test coverage (5 Core Test Methods covering essential scenarios).

**Project Structure**

**pps.solid.fmn (namespace)**

* + IMissingNumberFinder.cs # Core interface
  + LinqMissingNumberFinder.cs # LINQ implementation
  + HashSetMissingNumberFinder.cs # HashSet implementation
  + XorMissingNumberFinder.cs # XOR implementation
  + SumMissingNumberFinder.cs # Sum formula implementation
  + Program.cs # Main application & factory
  + Tests
    - MissingNumberFinderTests.cs # Unit tests
    - 20 Total Assertions (5 tests × 4 algorithms)
    - Edge Cases: Null input, empty arrays, single elements
    - Performance Verification: Large dataset handling

**Technical Specifications**

* **Framework**: .NET Framework 4.8.1
* **IDE**: Visual Studio 2019 compatible
* **Language**: C# 7.3
* **Testing**: MSTest Framework
* **Namespace**: pps.solid.fmn

**GitHub location**

<https://github.com/indigovn/SolidPrincipleImplementation>