

# Problem 258: Test Coverage

Difficulty: Medium

Author: Michael Warner, Denver, Colorado, United States

Originally Published: Code Quest 2025

## Problem Background

At Lockheed Martin, developing a good software product doesn't just require writing good code; it also needs good design and good testing. A group of people may all have various ideas about what some piece of software should do and how it should work. Once they come to agreement on those details, they eventually get documented as requirements in a formal document called a Software Requirements Specification. Each requirement is carefully written to avoid any confusion and receives a unique identifier to make it easier to reference.

Before a system gets delivered – regardless of whether it's a calculator program or the software that drives a massive rocket's launch system – every subsystem needs to be thoroughly tested to ensure that it meets the requirements. Otherwise, you might wind up with a giant explosion or merely a billion-dollar paperweight. To ensure that every requirement gets tested and there is nobody saying "oops, must have forgot about that one" 2 minutes after launch, the test procedures and their associated requirements are all cross-referenced in a big spreadsheet called the Requirements Traceability Matrix.

## Problem Description

Requirements for a launch computer on a rocket may look something like this:

- LC-001: The launch computer shall stop the countdown if it has not received GO\_FOR\_LAUNCH status from the control tower when the countdown time reaches T-00:00:10.
- LC-002: The launch computer shall initiate the launch sequence at T-00:00:00 if it has received GO\_FOR\_LAUNCH status from the control tower and its internal diagnostic status indicates no errors.
- LC-003: When the launch sequence has been initiated, the launch computer shall command the engines to ignite.

...And so on, and so on. As you can see, each requirement has a unique identifier, made up of a prefix and a number. In this example, those identifiers are LC-001, LC-002, LC-003. As noted above, each of these requirements must be tested using a defined procedure. These test procedures each have their own identifiers: LCTP-001, LCTP-002, LCTP-003, and so on.

The Requirements Traceability Matrix indicates which test procedure(s) test which requirement(s). A single test procedure might test multiple requirements. On the other hand, there might be multiple

test procedures needed to test a single requirement. If a requirement number matches a test procedure identifier, it is only a coincidence.

The Requirements Traceability Matrix for multiple subsystems will be given as input. Your program must read in the list of requirement identifiers and the list of requirements covered by each test procedure for the subsystem and ensure that every requirement is covered by at least one test procedure. If any requirements are not covered by a test procedure, your program must print out those requirement identifiers. If every requirement is covered by a test procedure, your program must print out “FULL COVERAGE”.

## Sample Input

The first line of your program’s input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include:

- A line containing a positive integer, **N**, indicating the number of entries in the Requirements Traceability Matrix
- A line containing a list of requirement identifiers separated by commas. Each requirement identifier will contain a prefix of letters and/or dashes, followed by a single integer which may include leading zeroes. These numbers may be presented in any order and may not represent a contiguous set of numbers (requirements get added and removed all the time!).
- **N** lines each containing an entry in the Requirements Traceability Matrix. Each entry will include the following information, separated by commas:
  - A test identifier, which will contain a prefix of letters and/or dashes, followed by a single integer which may include leading zeroes. The prefix will not match that of any requirement in the same test case. Numbers may be presented in any order and may not represent a contiguous set of numbers.
  - One or more requirement identifiers from the list previously given.

```
2
1
LC-001,LC-003,LC-002,LC-015,LC-105
LCTP-002,LC-003,LC-001
3
MC-515,MC-616,MC-007,MC-009,MC-112
MCTP-001,MC-007,MC-112
MCTP-005,MC-515
MCTP-008,MC-616,MC-009
```

## Sample Output

For each test case, your program must print the requirement identifiers that are not covered by a test procedure, separated by commas. Print requirement identifiers in default string ordering (that is,

sorted according to the ordering given by the US ASCII table in the reference materials). If all requirements are covered by a test procedure, your program must print FULL COVERAGE.

**LC-002, LC-015, LC-105**  
**FULL COVERAGE**