

Good Testing Must Consider Both Purpose and Structure

Full code coverage is **just a starting point**.

In fact, you should notice that

- one of our tests (“0 0 0”)
- exposes a bug
- in a statement that was already covered
- by another test (“1 0 0”).

In general, good testing requires that one **think carefully about the purpose** of the code **as well as the structure** of the code.

***** So Easy that a Computer Can Do It

Full code coverage is easy to explain.

Finding tests to cover more statements means solving some equations.

Computers are good at that (well ... pretty good).

The automatic programming feedback tool uses this approach to try to find bugs in your code:

- generate tests to cover everything (if possible),
- then compare your program's results with a “gold” program (written by a professor or TA).