1.

// MylibraryTests.swift

```
var isLuckyNumber: Bool?
```

The Variable isLuckyNumber stores a Boolean value.

```
myLibrary.isLucky(number, completion: { lucky in
    isLuckyNumber = lucky
    expectation.fulfill()
})
```

isLucky function from //MyLibrary.swift is called.

// Test Pass / Fail check

1.

```
XCTAssertNotNil(isLuckyNumber)
```

Checks if the variable isLuckyNumber is NULL or NOT.

The Assertion fails if the variable isLuckyNumber is NULL.

2.

```
XCTAssert(isLuckyNumber == true)
```

After the first assertion is passed, this checks whether the variable isLuckyNumber is a True or False value.

The Assertion is passed when the variable isLuckyNumber is true and fails otherwise.

// MyLibrary.swift isLucky() analysis:

1. Without bugfix:

```
public func isLucky(_ number: Int, completion: @escaping (Bool?)
-> Void) {
        // Check the simple case first: 3, 5 and 8 are
automatically lucky.
        if number == 3 || number == 5 || number == 8 {
            completion(true)
        }
```

When the function isLucky is called from //MtLibraryTests.swift , the number 8 is sent to the isLucky() however after the if block, the function is not returning any value back to the function call where the isLuckyNumber variable holds the NULL value failing at :

```
XCTAssertNotNil(isLuckyNumber)
```

2. Bugfix:

```
public func isLucky(_ number: Int, completion: @escaping (Bool?)
-> Void) {
        // Check the simple case first: 3, 5 and 8 are
automatically lucky.
        if number == 3 || number == 5 || number == 8 {
            completion(true)
            return
        }
```

Adding a return statement here sends the boolean TRUE value to the function call and replaces the variable isLuckyNumber is overridden with the value returned by the function isLucky(). This clears the first assertion

```
XCTAssertNotNil(isLuckyNumber)
```

because, the variable isLuckyNumber has a value in it.

The second assertion

```
XCTAssert(isLuckyNumber == true)
```

Is passed because the variable isLuckyNumber holds the boolean value TRUE.

In addition, when the return statement is not included, the true is overridden with false because 39 is returned which is not in the if condition. Therefore the assertion fails