User Story:

1.As a user interested in checking individual numbers, I want to input a specific number into the Armstrong number checker, so that I can quickly verify whether that particular number is an Armstrong number or not.

2.As a developer, I want to optimize the Armstrong number checker algorithm to efficiently handle large numbers, so that users can confidently check the Armstrong property for a wide range of numerical inputs.

3.As a quality assurance tester, I want to create test cases that cover various scenarios, including positive and negative numbers, to ensure the Armstrong number checker functions accurately and reliably, meeting the requirements specified in the user stories.

TEST CASE DOCUMENTATION

Test method name	**Test description**	**Input**	**Expected output**	**Actual output**
TestCase1:: Test_Armstrong_Number	Check whether the method identifies the known armstrong number as valid	153	TRUE	TRUE
TestCase2: Test_Armstrong_Number	Check whether the method identifies the known armstrong number as valid	370	TRUE	TRUE
TestCase3:: Test_Non_Armstrong_Number	Check whether the method identifies the known non armstrong number as invalid	123	FALSE	FALSE
TestCase4: Test_Non_Armstrong_Number	Check whether the method identifies the known non armstrong number as invalid	1635	FALSE	FALSE
TestCase5:: Test_Armstrong_Number_With_ Large_Number	Check whether the method handles large numbers correctly	9474	TRUE	TRUE
TestCase6:: Test_Armstrong_Number_With_ Large_Number	Check whether the method handles large numbers correctly	8208	TRUE	TRUE
TestCase7:: Test_Armstrong_Number_With_ Negative_Number	Check whether the method handles negative numbers correctly	-153	FALSE	FALSE
TestCase8:: Test_Armstrong_Number_With_ Negative_Number	Check whether the method handles negative numbers correctly	-203	FALSE	FALSE