



CS6474: Software Testing Laboratory
(Spring 2023)

Bishwajit Prasad Gond
222CS3113

Master of Technology
222cs3113@nitrkl.ac.in

Department of Computer Science & Engineering
NIT, Rourkela

March 5, 2023

Contents

1	Jumble	1
1.1	Write a program to generate a Factorial of numbers (where stack length should be at 3 (max)). The numbers should be 5, 3, 8, and 15.	2
1.2	Write a program to generate Fibonacci numbers.	3
1.3	Write a program that performs sorting of a group of integer values using the quick sort technique.	4
1.4	Write a program that accepts elements of a matrix and displays its transpose. .	5
1.5	Write a program to add two matrices and display the sum matrix.	6
1.6	Write a program to Print Prime Numbers from 1 to 100 using Scanner Class and For Loop.	7
1.7	Write a program to generate a palindrome of numbers.	8
1.8	Write a program to find out the sum of two arrays.	9
1.9	Write a program to check whether the number is even or odd.	10
1.10	Write a program for binary to hexadecimal conversion.	11

1 Jumble

Jumble is a mutation testing tool which changes the Java code at byte code level. It runs a unit test case and applies certain set of mutations.

Passed Mutations: Changes in code which do not affect the final output

Failed Mutations: Changes in code which change the final output

Score : It is the percentage of passed mutations out of the total mutations. It is used to assess the validity of our test suite.

Jumble is a class level mutation testing tool that works in conjunction with JUnit. The purpose of mutation testing is to provide a measure of the effectiveness of test cases. A single mutation is performed on the code to be tested, the corresponding test cases are then executed. If the modified code fails the tests, then this increases confidence in the tests. Conversely, if the modified code passes the tests this indicates a testing deficiency.

Example Here is some example Jumble output for a Java class called "Foo", which has some JUnit tests in a class called "FooTest".

Jumble starts by running the unit tests (in FooTest.class) on the unmodified Foo class to check that they all pass, and to measure the time taken by each test. Then it will mutate Foo in various ways and run the tests again to see if they detect the mutation. It continues this process until all mutations of Foo have been tried.

Installation and Usage

Release 1.3.0 of Jumble is available as an Eclipse plugin. Just add <http://jumble.sourceforge.net/update/> to Eclipse as an update site, and you will be able to install Jumble. Then you should be able to right-click on any class and do Jumble Mutation Tester / Analyse tests of this class to mutate the class and analyse its tests. Note that you click on your source code class, NOT on the JUnit tests.

For command line usage, just download `jumble_1.x.y.jar` via the Sourceforge Files release system, and run it with `java -jar jumble_1.x.y.jar -help` to see the help message.

Solve the below programs in Java

1.1 Write a program to generate a Factorial of numbers (where stack length should be at 3 (max)). The numbers should be 5, 3, 8, and 15.

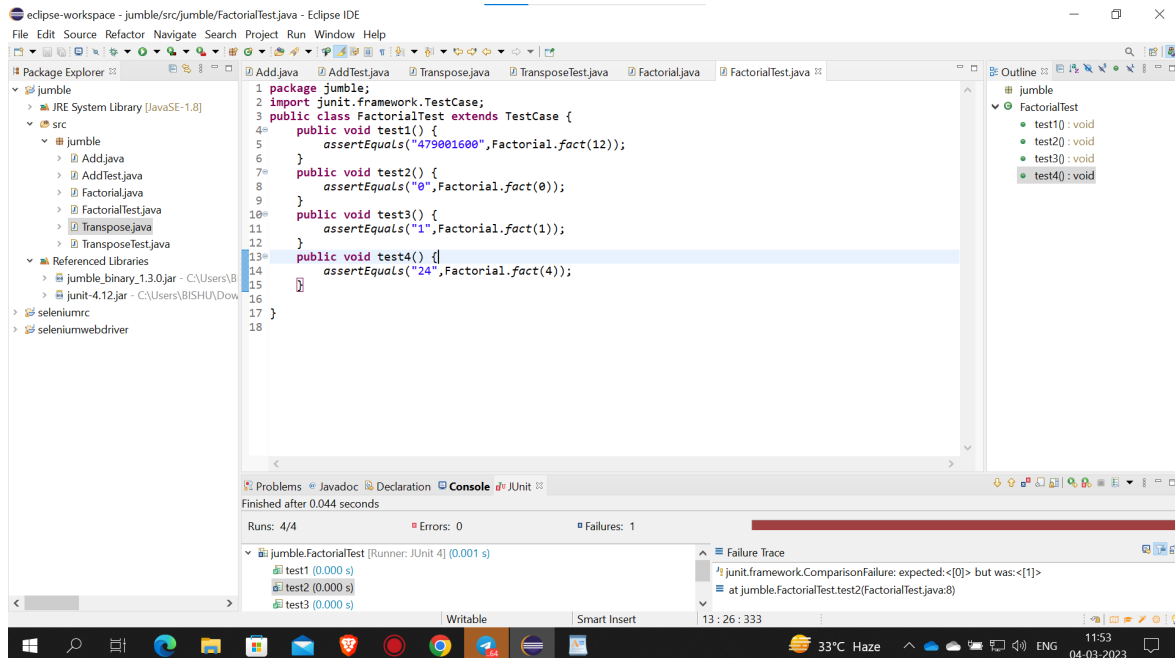


Figure 1: Jumble Test Case

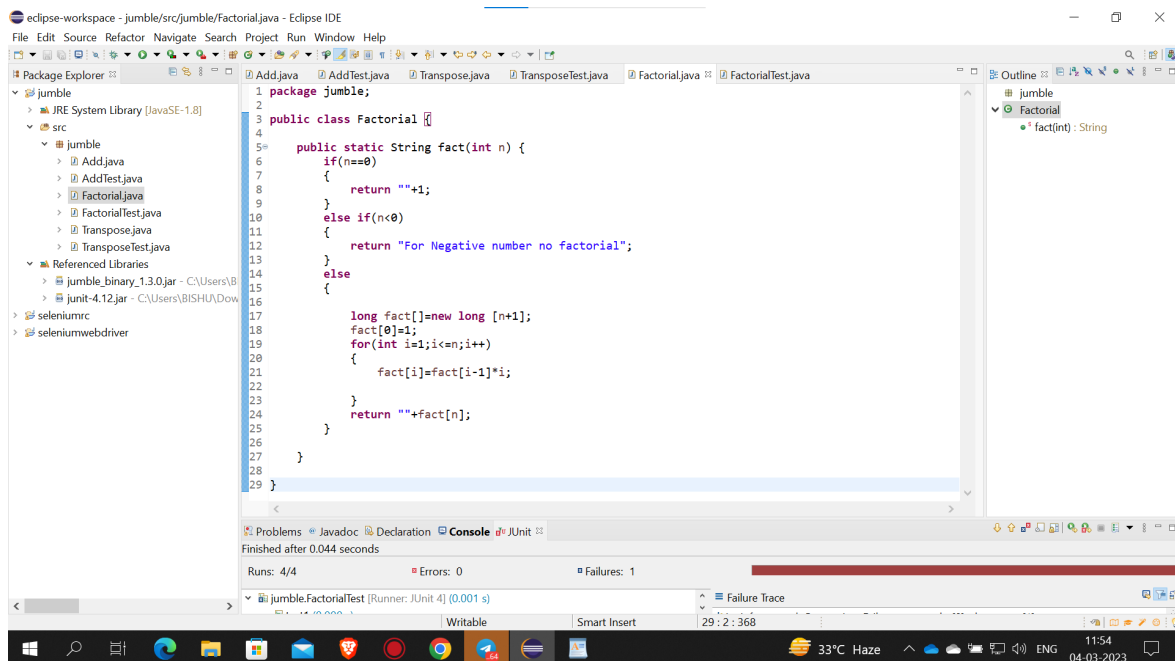


Figure 2: Factorial Code

1.2 Write a program to generate Fibonacci numbers.

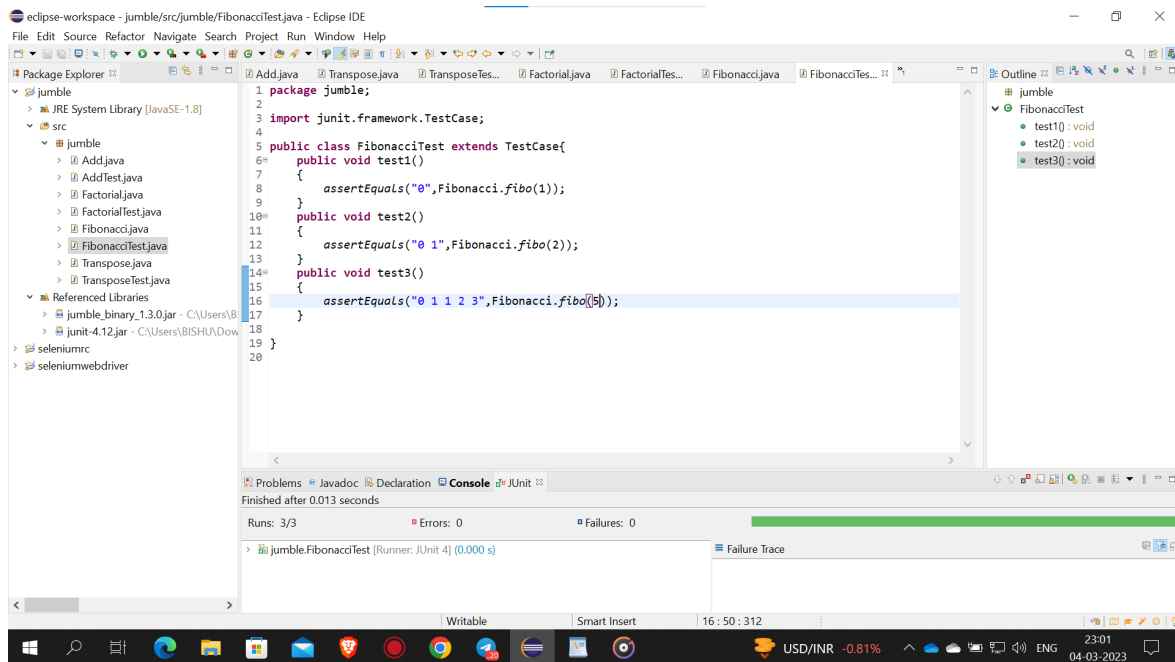


Figure 3: Jumble Test Case

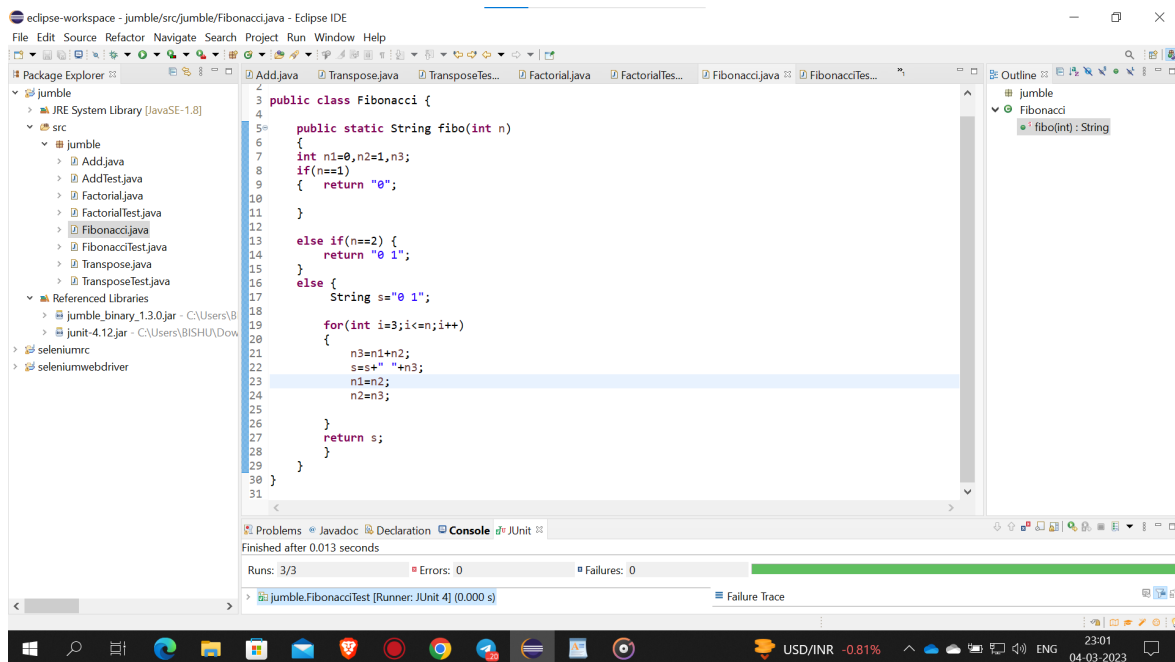


Figure 4: Fibonacci Code

1.3 Write a program that performs sorting of a group of integer values using the quick sort technique.

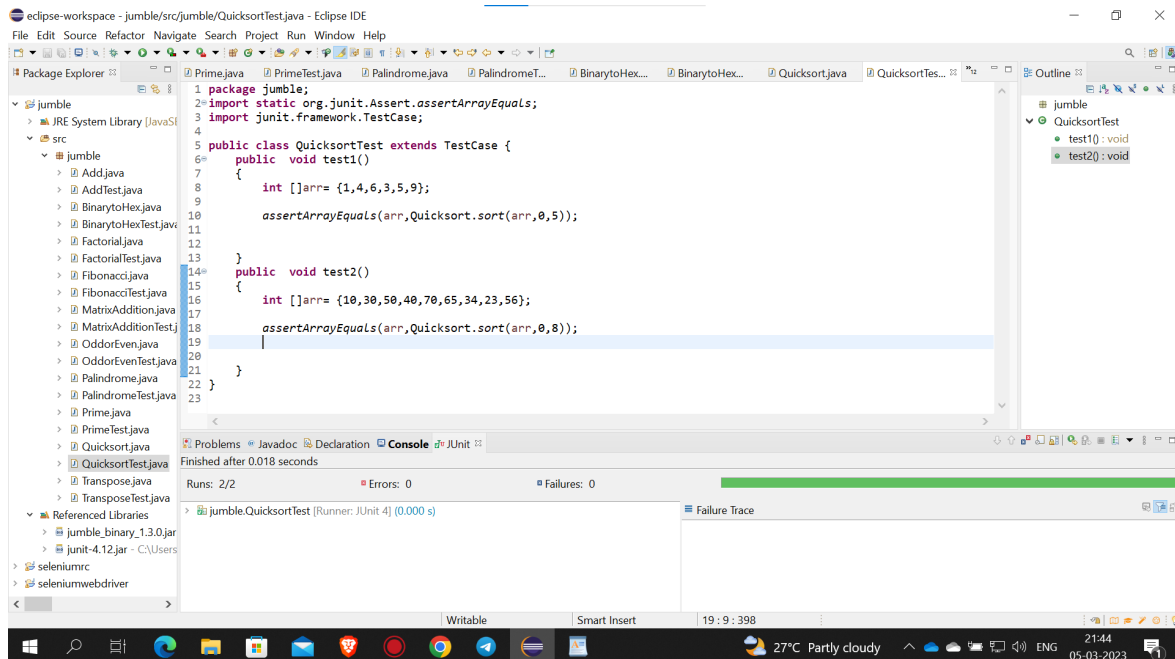


Figure 5: Jumble Test Case

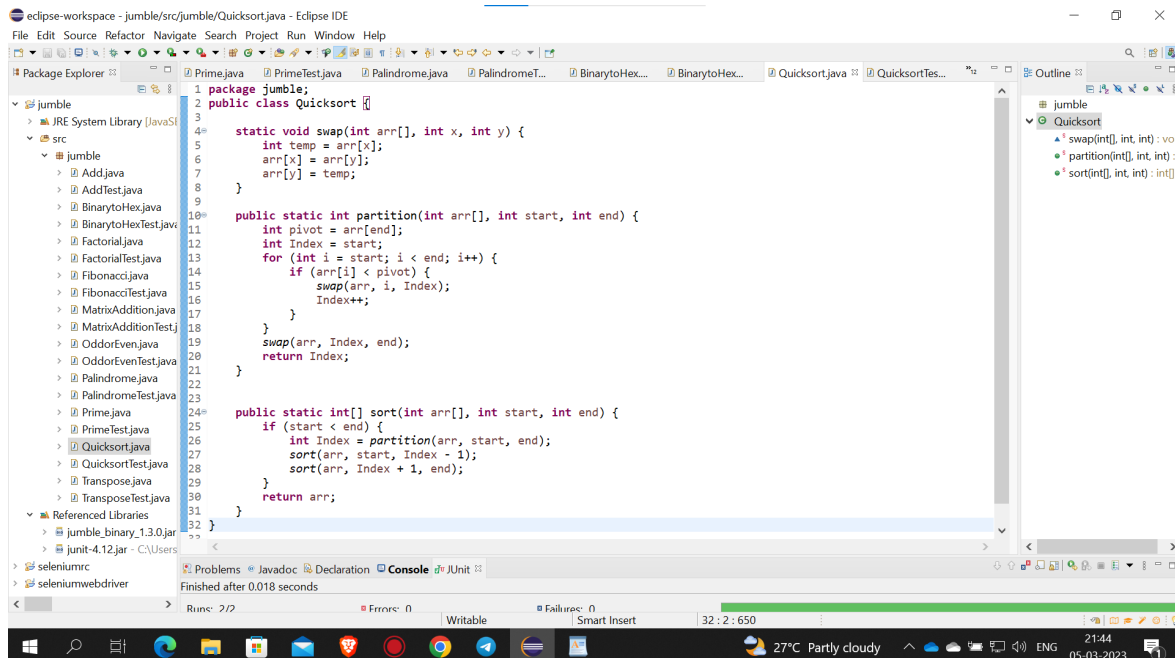


Figure 6: Quicksort Code

1.4 Write a program that accepts elements of a matrix and displays its transpose.

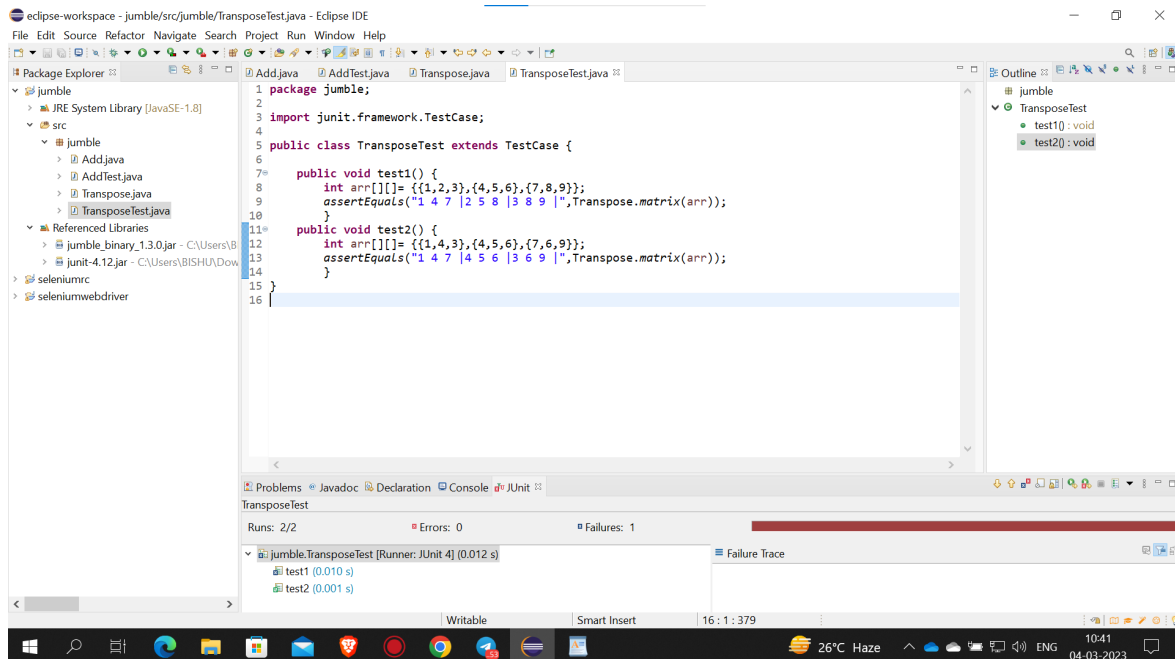


Figure 7: Jumble Test Case

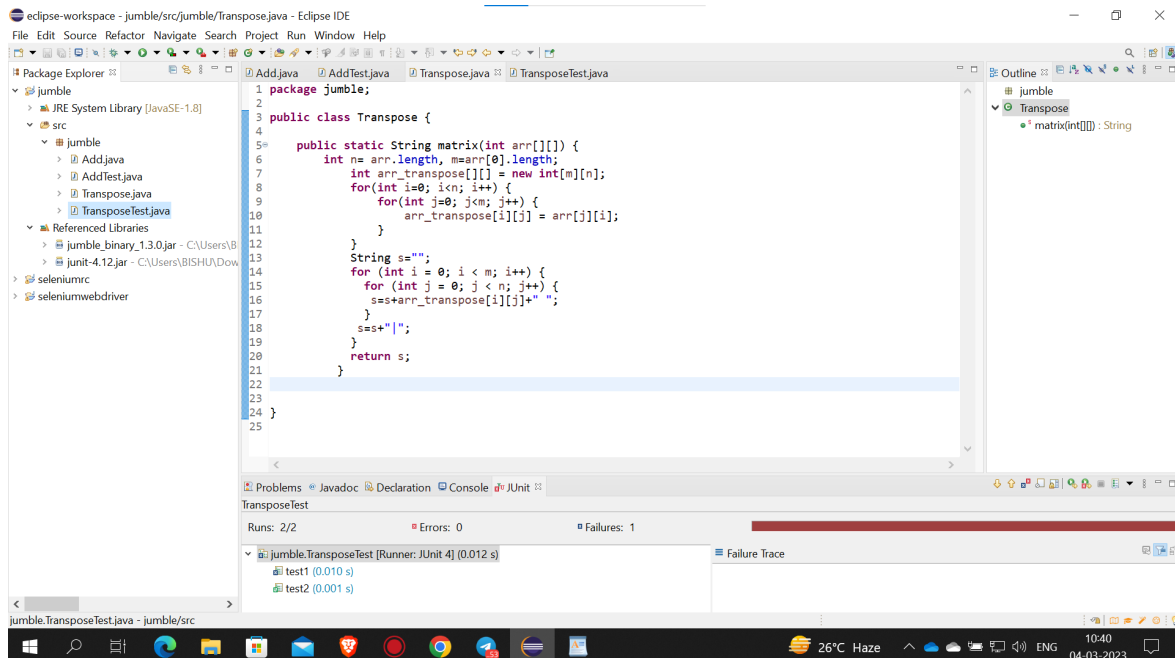


Figure 8: Matrix Transpose Code

1.5 Write a program to add two matrices and display the sum matrix.

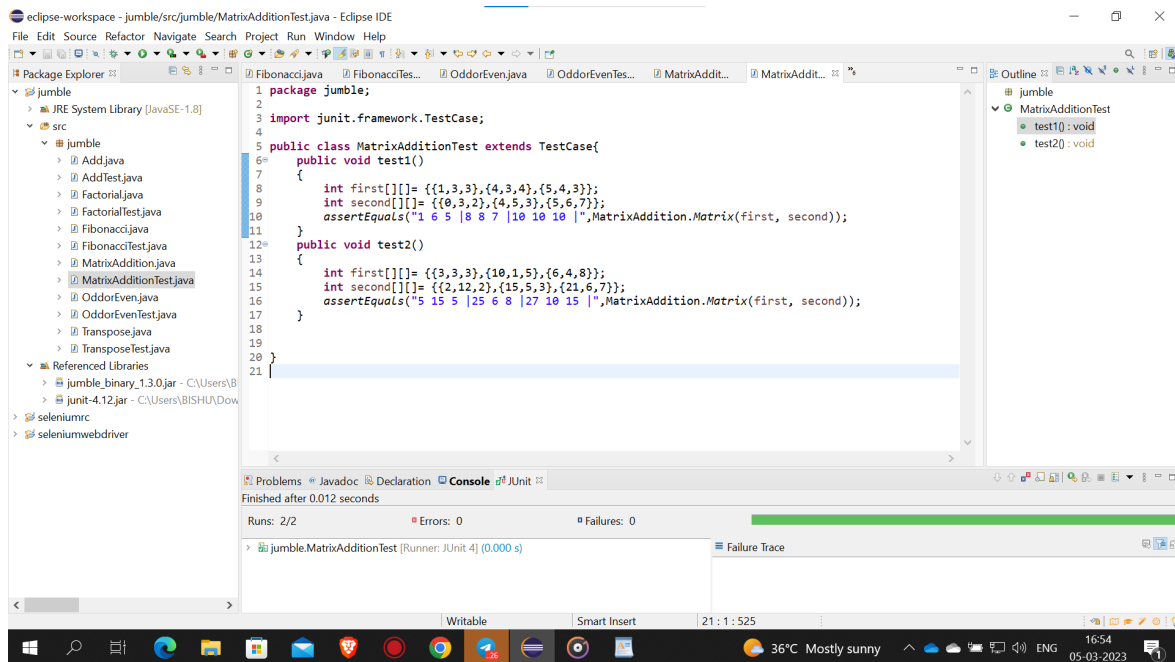


Figure 9: Jumble Test Case

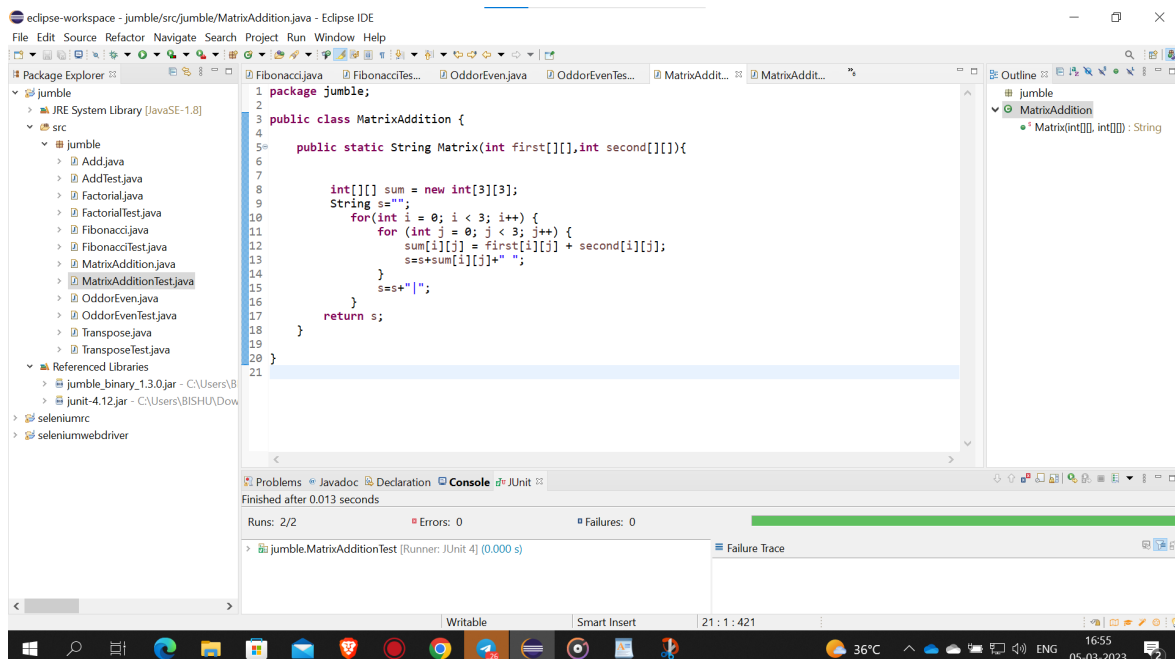


Figure 10: Matrix Addition Code

1.6 Write a program to Print Prime Numbers from 1 to 100 using Scanner Class and For Loop.

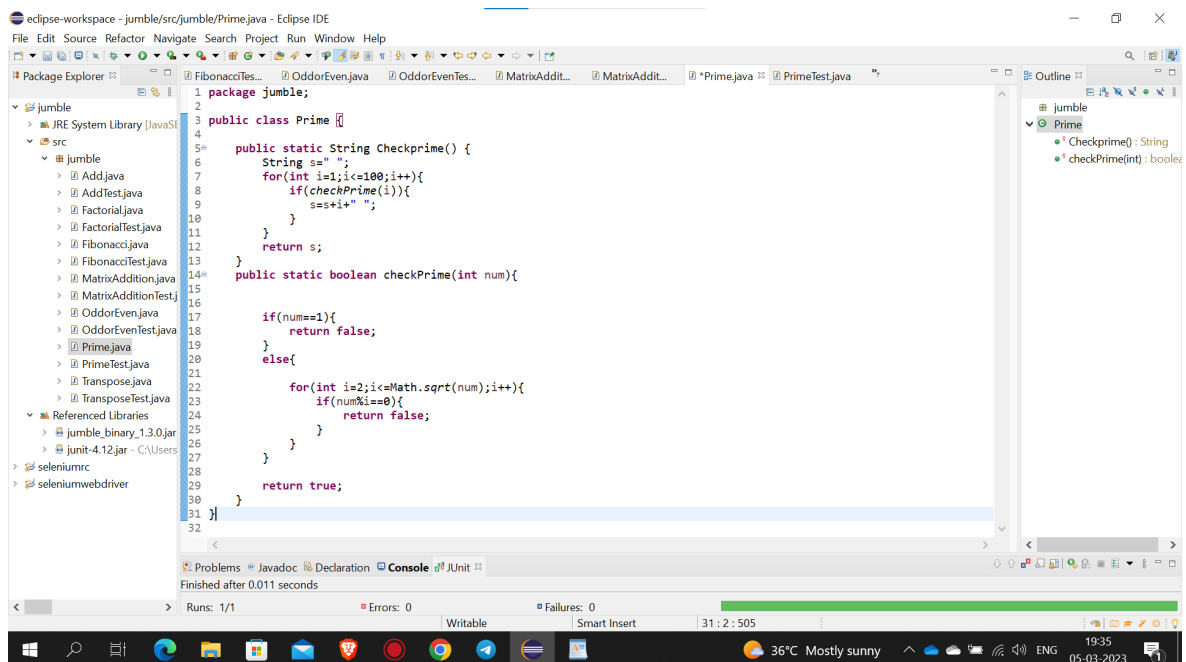


Figure 11: Jumble Test Case

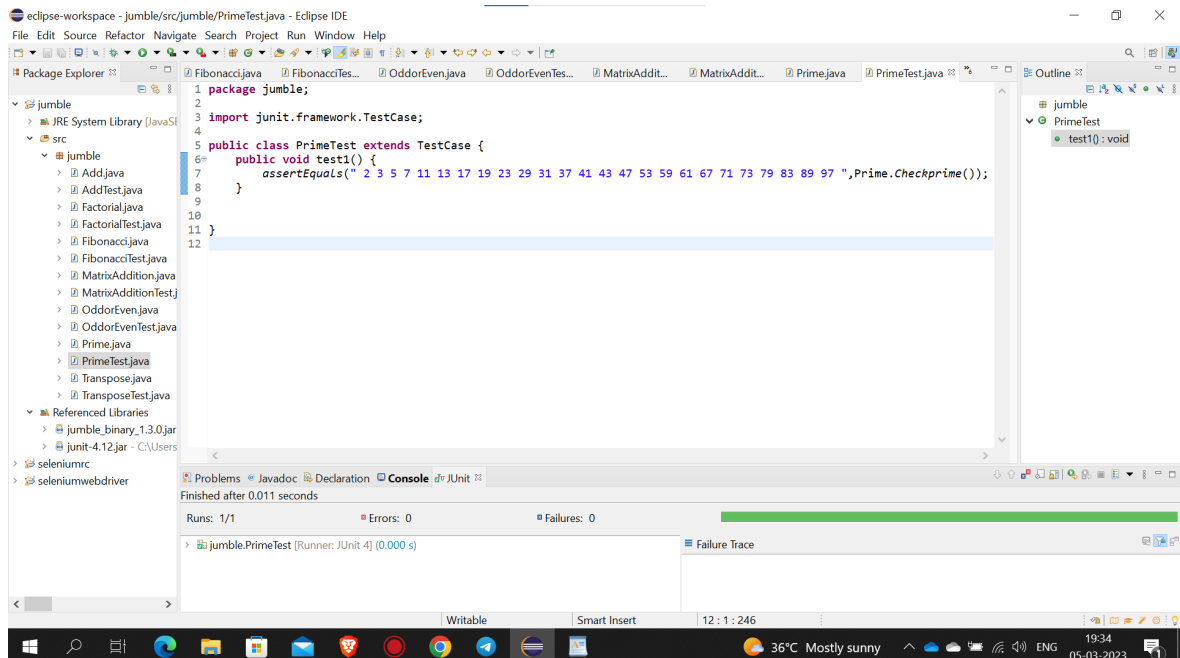


Figure 12: Prime Code

1.7 Write a program to generate a palindrome of numbers.

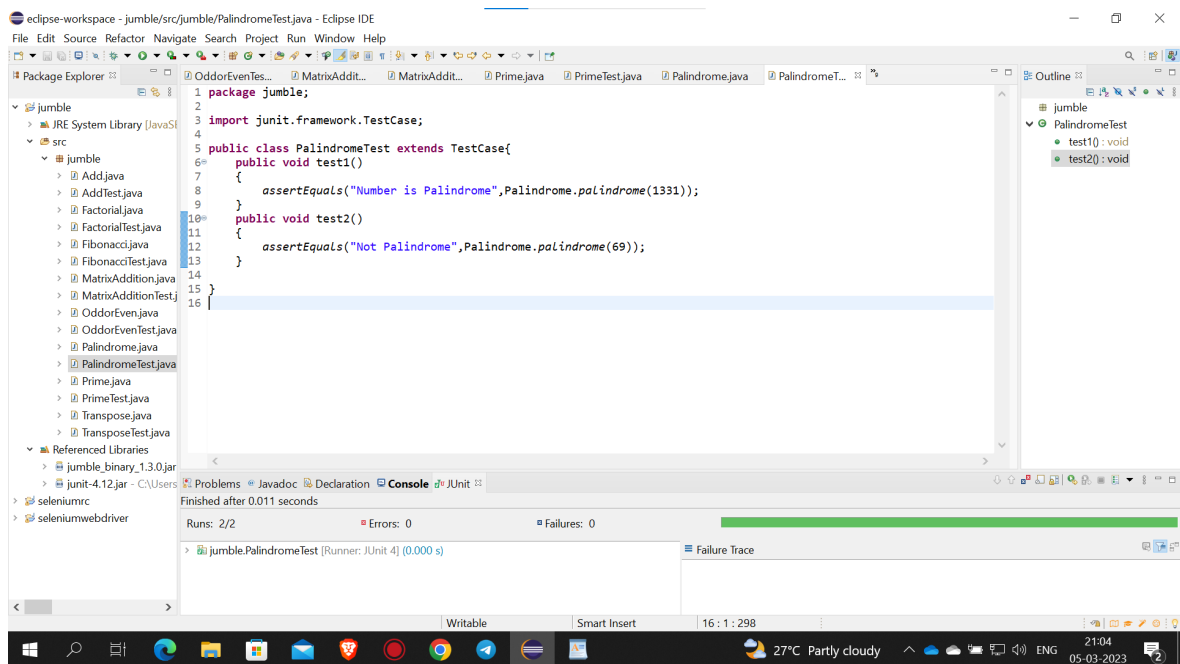


Figure 13: Jumble Test Case

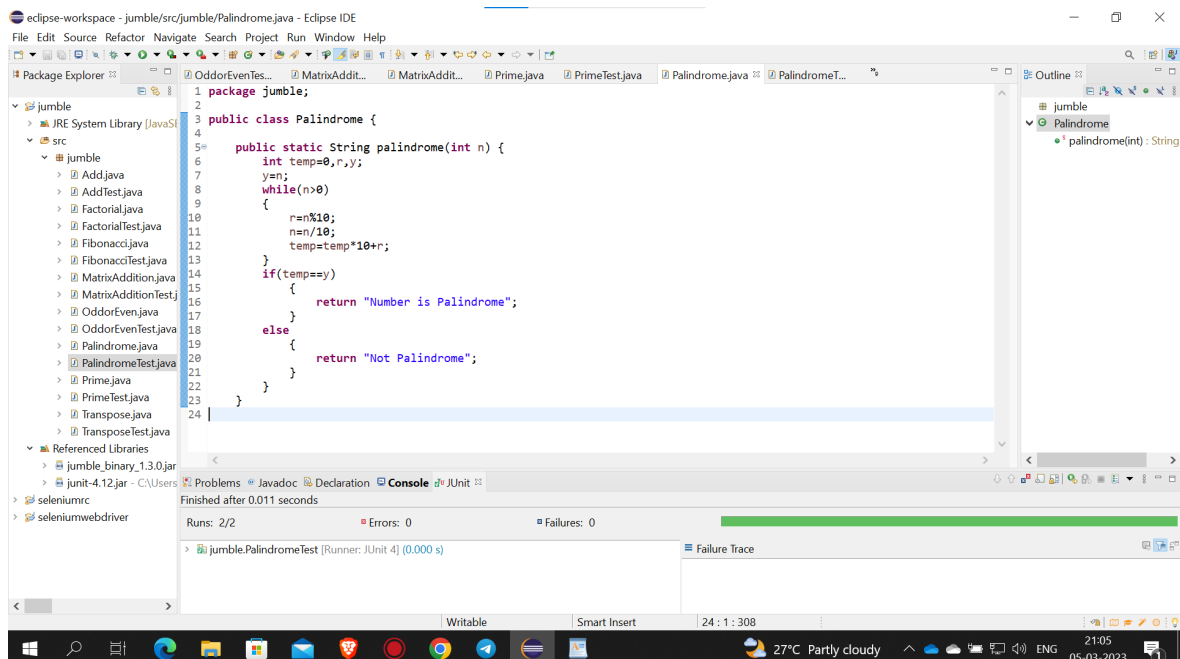


Figure 14: Palindrome Code

1.8 Write a program to find out the sum of two arrays.

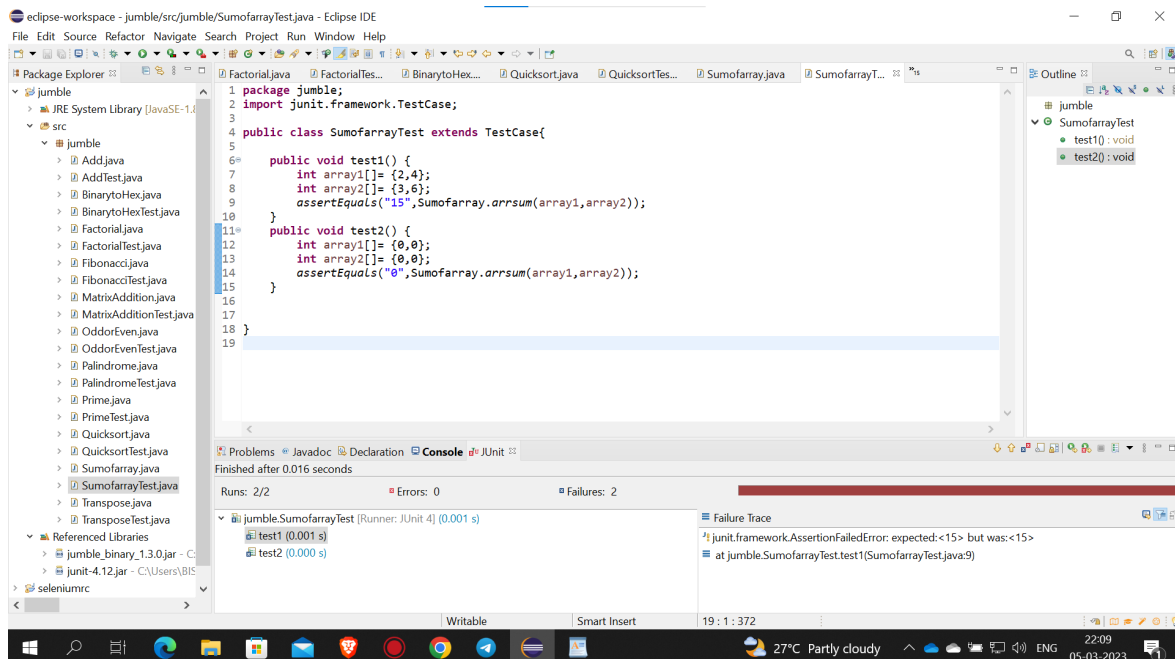


Figure 15: Jumble Test Case

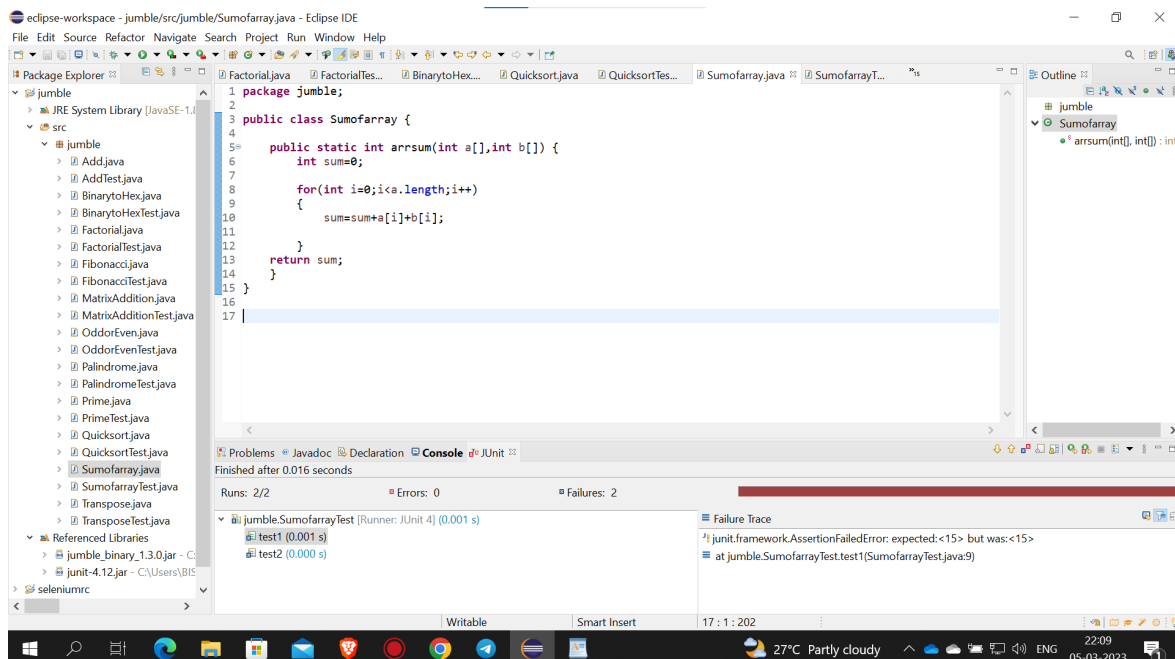


Figure 16: Sum of Array Code

1.9 Write a program to check whether the number is even or odd.

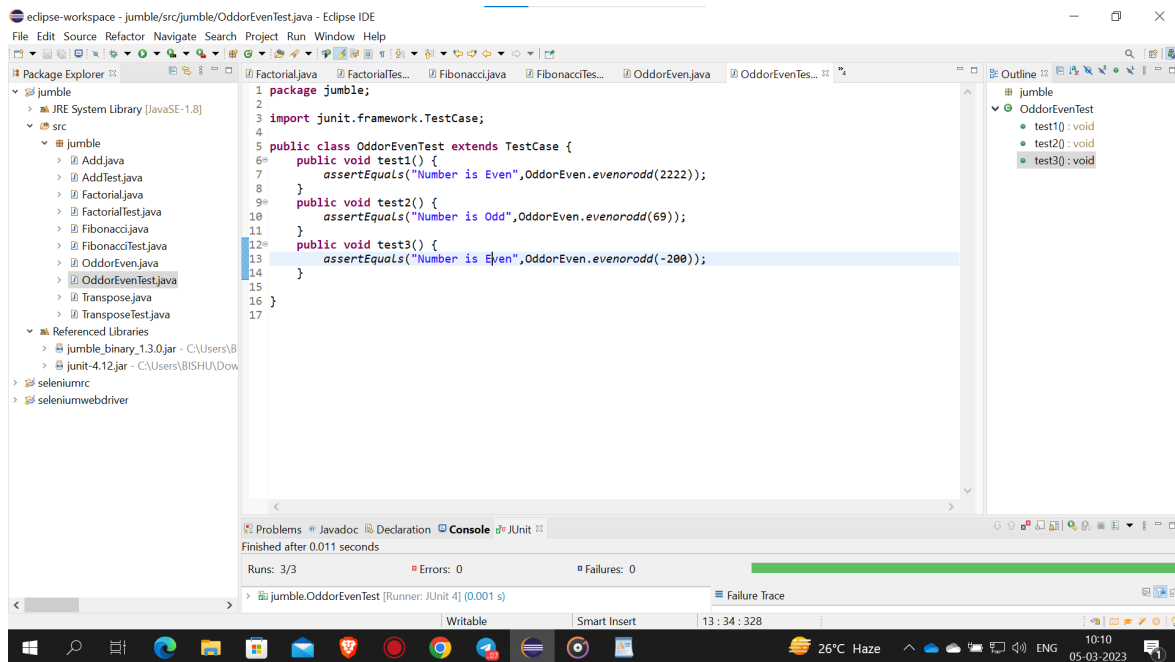


Figure 17: Jumble Test Case

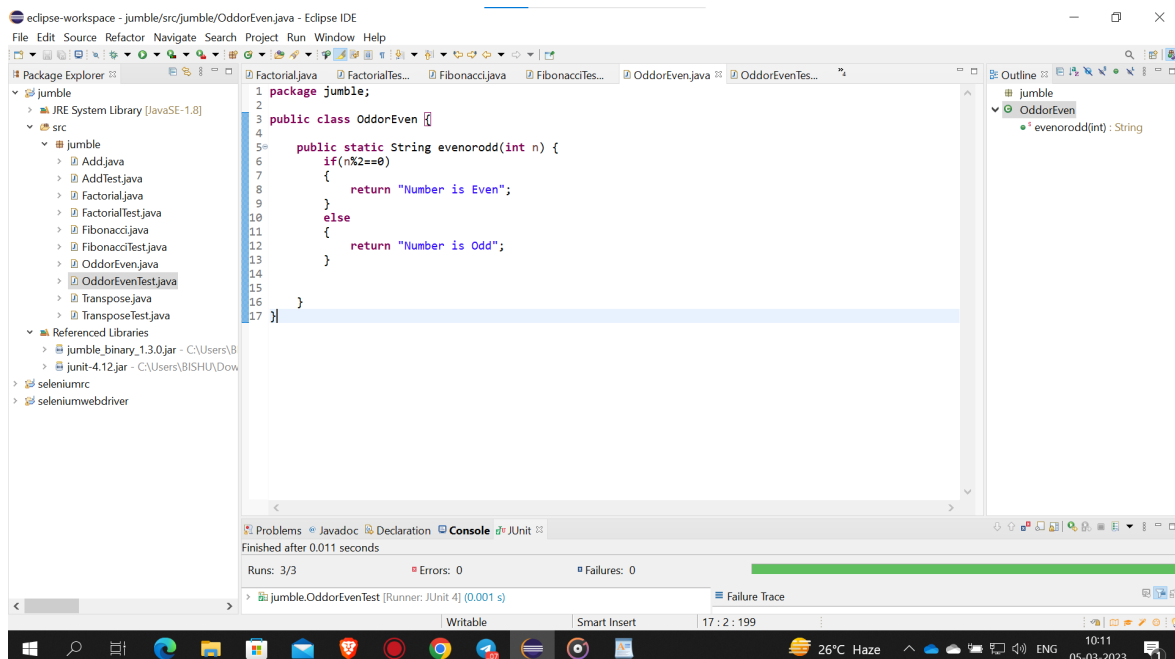


Figure 18: Odd or Even Code

1.10 Write a program for binary to hexadecimal conversion.

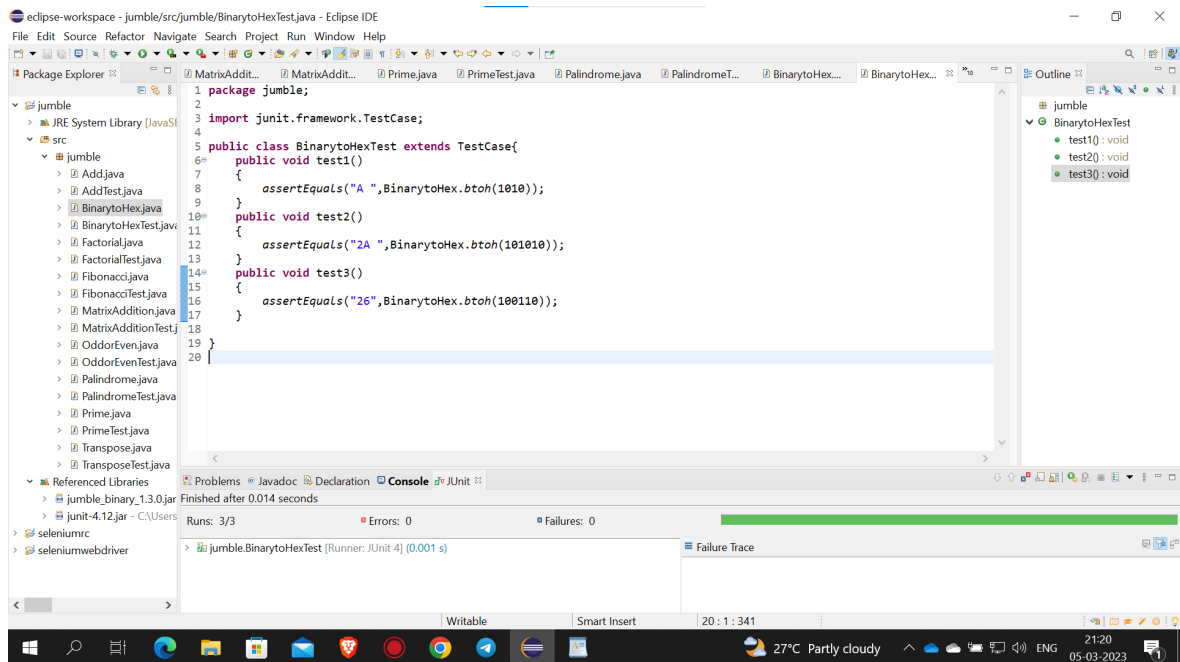


Figure 19: Jumble Test Case

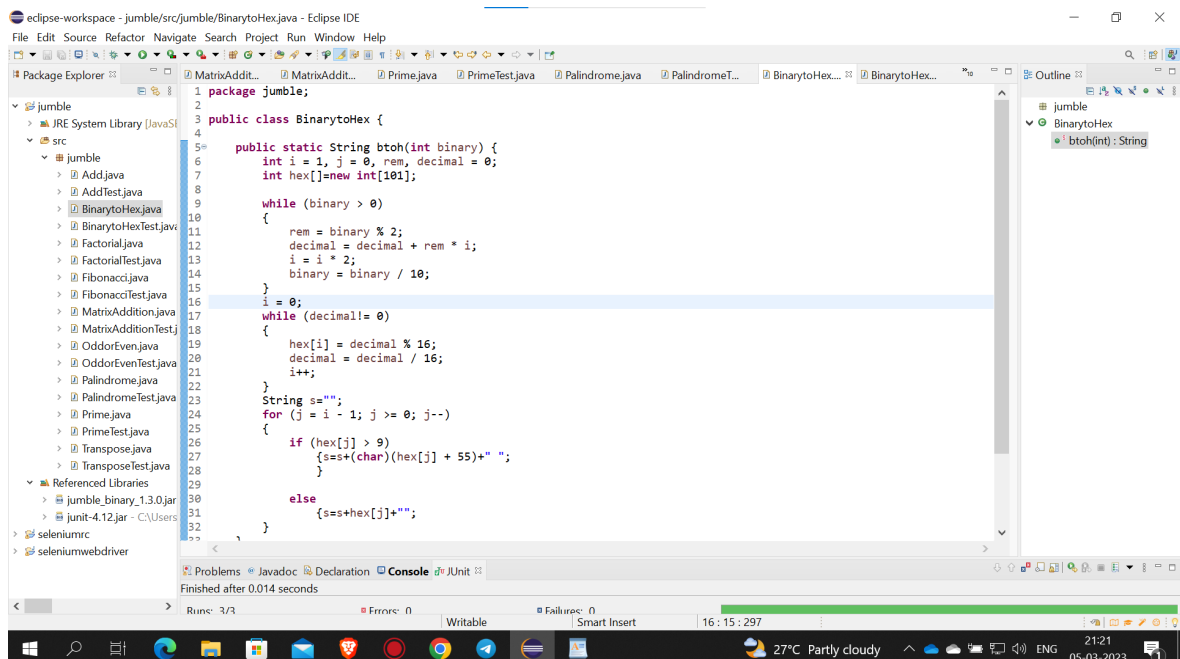


Figure 20: Binary to Hexadecimal Code