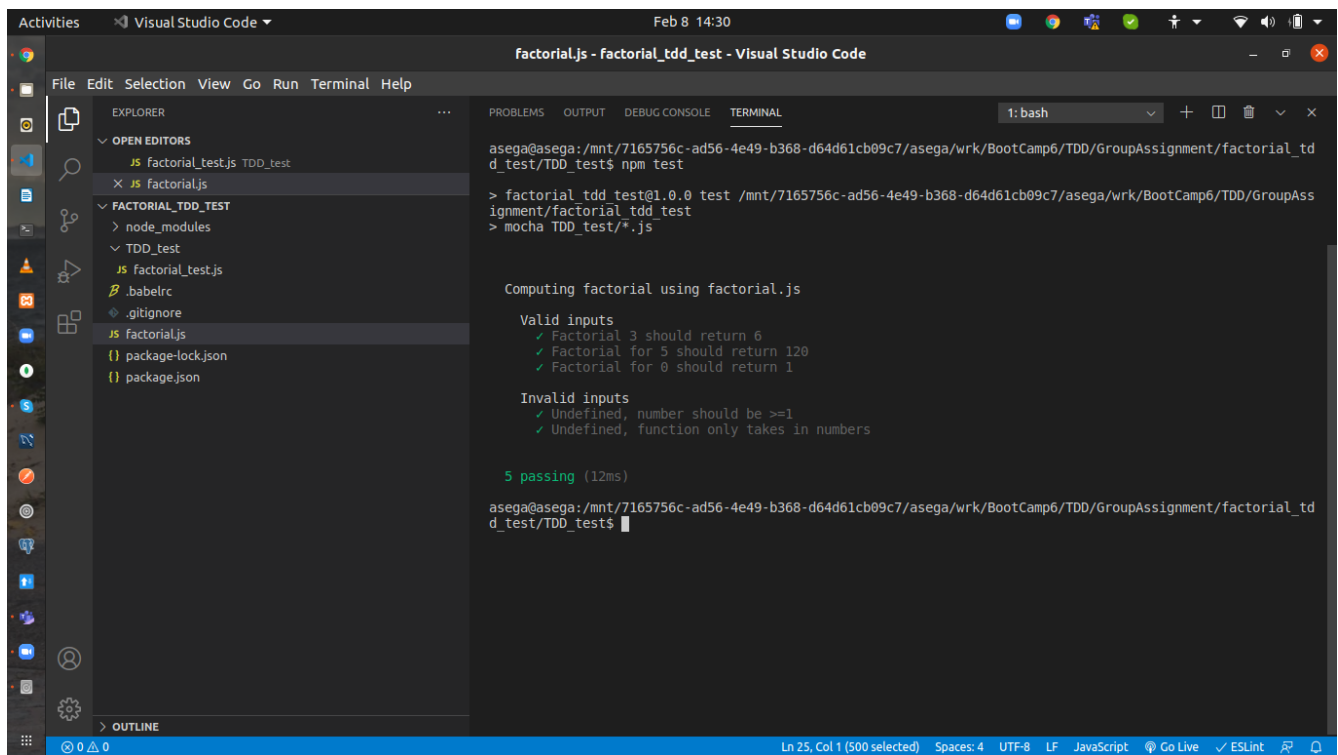


Factorial Iterative approach: Success TDD Tests



The screenshot shows the Visual Studio Code interface with the Explorer view on the left displaying the project structure. The file explorer shows a folder named 'factorial_tdd_test' containing files like 'factorial_test.js', 'node_modules', 'TDD_test', 'factorial_test.js', '.babelrc', '.gitignore', 'factorial.js', 'package-lock.json', and 'package.json'. The main editor area shows the 'factorial_test.js' file with the following code:

```
function factorial(n) {
  let result = 1;
  for (let i = 1; i <= n; i++) {
    result *= i;
  }
  return result;
}
```

The terminal window at the bottom shows the command 'npm test' being executed, which runs the tests defined in 'factorial_test.js'. The output of the tests is as follows:

```
Computing factorial using factorial.js

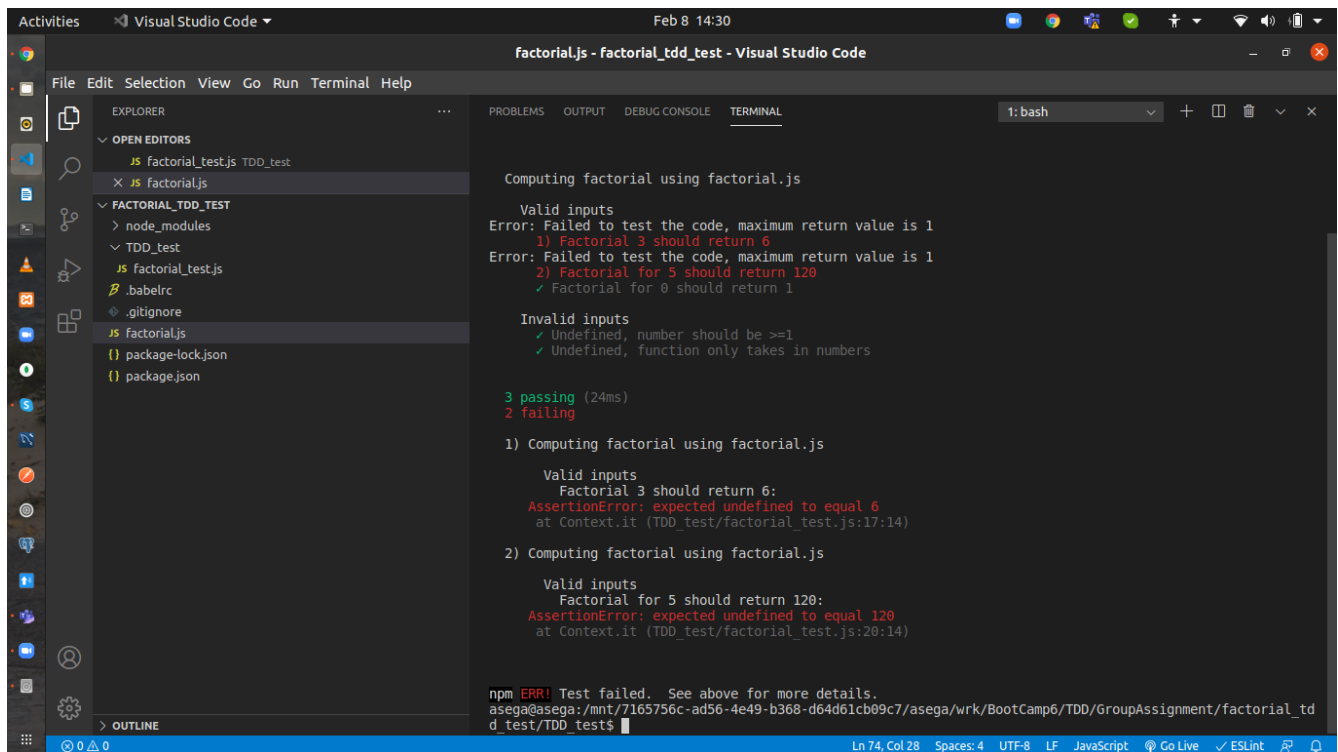
Valid inputs
  ✓ Factorial 3 should return 6
  ✓ Factorial for 5 should return 120
  ✓ Factorial for 0 should return 1

Invalid inputs
  ✓ Undefined, number should be >=1
  ✓ Undefined, function only takes in numbers

5 passing (12ms)
```

The status bar at the bottom indicates the current line is 25, column 1, with 500 characters selected. The file encoding is UTF-8, and the file type is JavaScript.

Factorial Recursive approach: Success and Failed TDD Tests (throwing catch (err))



The screenshot shows the Visual Studio Code interface with the Explorer view on the left displaying the project structure. The file explorer shows a folder named 'factorial_tdd_test' containing files like 'factorial_test.js', 'node_modules', 'TDD_test', 'factorial_test.js', '.babelrc', '.gitignore', 'factorial.js', 'package-lock.json', and 'package.json'. The main editor area shows the 'factorial_test.js' file with the following code:

```
function factorial(n) {
  if (n < 1) {
    return 1;
  }
  return n * factorial(n - 1);
}
```

The terminal window at the bottom shows the command 'npm test' being executed, which runs the tests defined in 'factorial_test.js'. The output of the tests is as follows:

```
Computing factorial using factorial.js

Valid inputs
Error: Failed to test the code, maximum return value is 1
1) Factorial 3 should return 6
Error: Failed to test the code, maximum return value is 1
2) Factorial for 5 should return 120
  ✓ Factorial for 0 should return 1

Invalid inputs
  ✓ Undefined, number should be >=1
  ✓ Undefined, function only takes in numbers

3 passing (24ms)
2 failing

1) Computing factorial using factorial.js
   Valid inputs
     Factorial 3 should return 6:
     AssertionError: expected undefined to equal 6
     at Context.it (TDD_test/factorial_test.js:17:14)

2) Computing factorial using factorial.js
   Valid inputs
     Factorial for 5 should return 120:
     AssertionError: expected undefined to equal 120
     at Context.it (TDD_test/factorial_test.js:20:14)

npm ERR! Test failed. See above for more details.
asega@asega:/mnt/7165756c-ad56-4e49-b368-d64d61cb09c7/asega/wrk/BootCamp6/TDD/GroupAssignment/factorial_tdd_test/TDD_test$
```

The status bar at the bottom indicates the current line is 74, column 28, with 4 spaces. The file encoding is UTF-8, and the file type is JavaScript.