.

Lesson 02 Demo 01

Finding the Missing Number

Objective: To design an algorithm that identifies a missing integer in a sequence, enhancing logical thinking and foundational algorithm skills

Tools required: Visual Studio Code and JavaScript

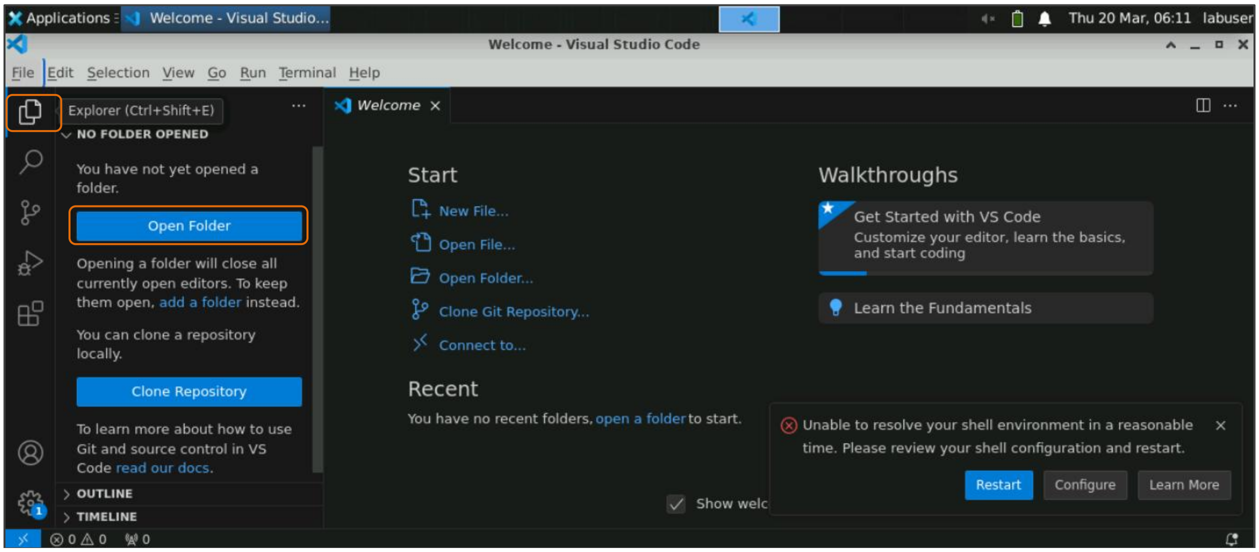
Prerequisites: None

Steps to be followed:

1. Create a working directory within the lab environment
2. Create a JavaScript file and execute it

Step 1: Create a working directory within the lab environment

1.1 Launch Visual Studio Code, click on the **Explorer** icon, and select **Open Folder**



You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

To learn more about how to use Git and source control in VS Code read our docs.

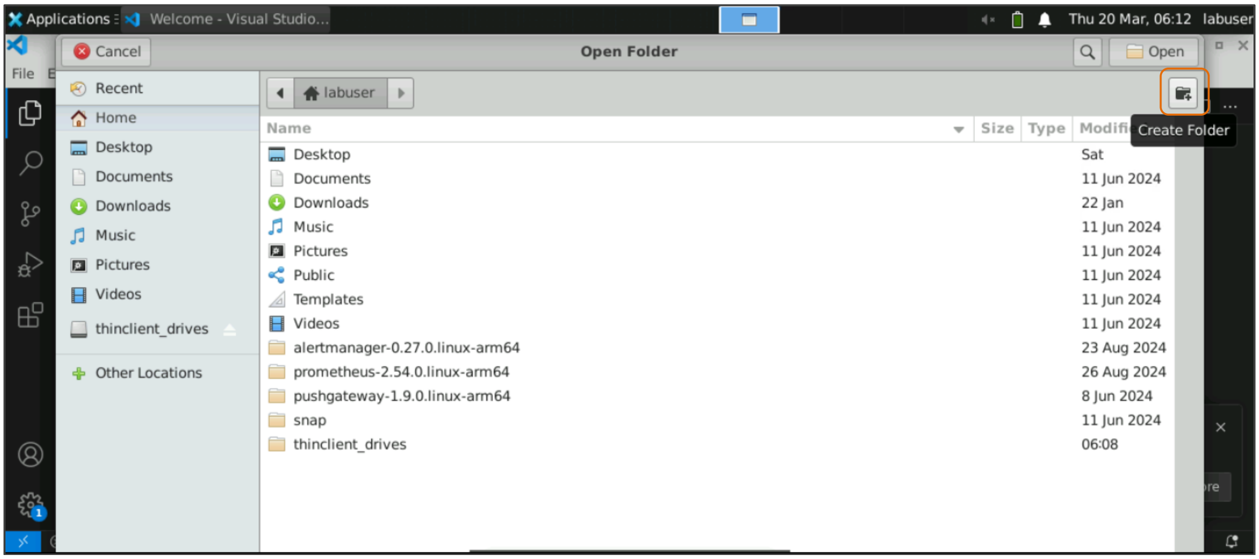
You have no recent folders, open a folder to start.

Unable to resolve your shell environment in a reasonable time. Please review your shell configuration and restart.

.

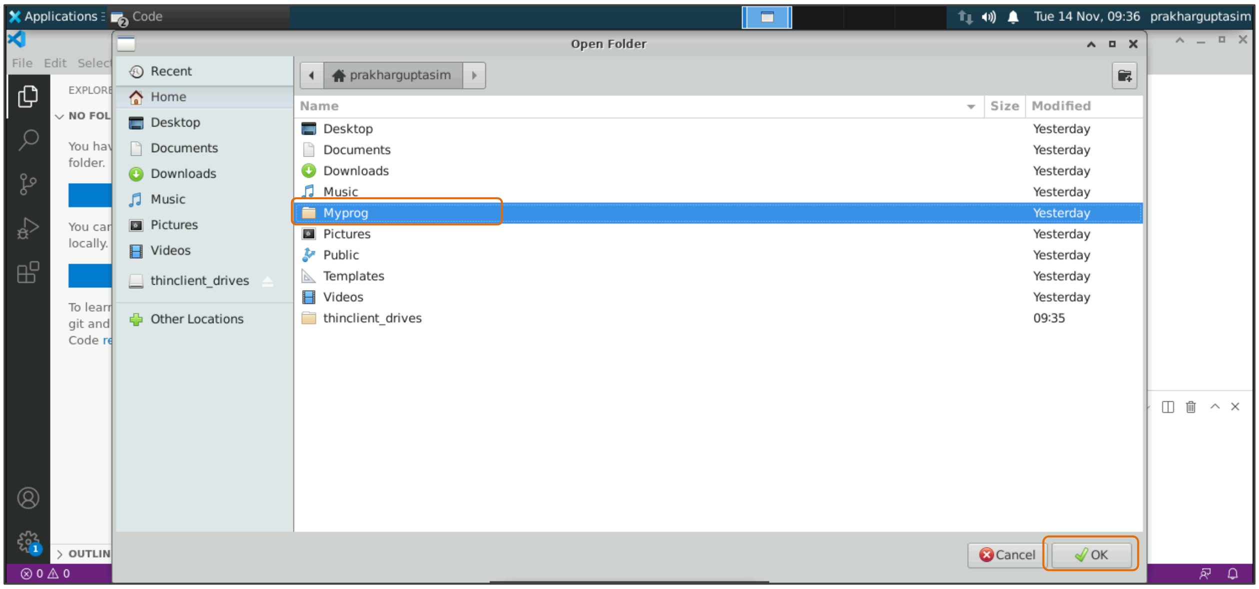
1.2 Create an arbitrary folder, which will be used as your working directory

1.2.1 Click on the **Create Folder** icon



Name	Size	Type	Modified
Desktop			Sat
Documents			11 Jun 2024
Downloads			22 Jan
Music			11 Jun 2024
Pictures			11 Jun 2024
Public			11 Jun 2024
Templates			11 Jun 2024
Videos			11 Jun 2024
alertmanager-0.27.0.linux-arm64			23 Aug 2024
prometheus-2.54.0.linux-arm64			26 Aug 2024
pushgateway-1.9.0.linux-arm64			8 Jun 2024
snap			11 Jun 2024
thinclient_drives			06:08

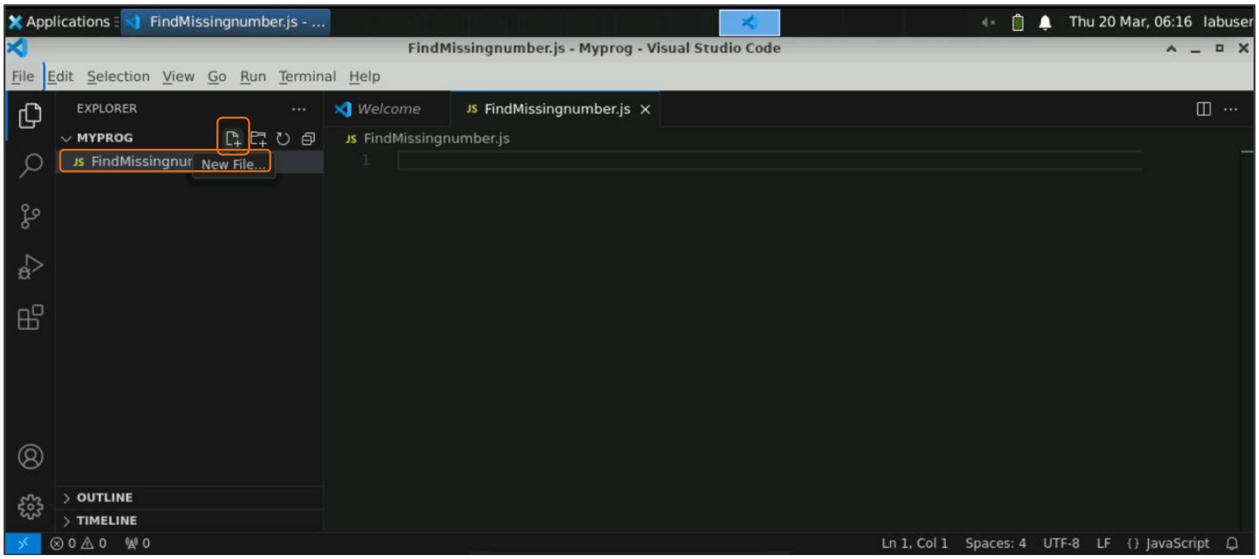
1.2.2 Create a folder named **Myprog** and then click **OK**



Name	Size	Modified
Desktop		Yesterday
Documents		Yesterday
Downloads		Yesterday
Music		Yesterday
Myprog		Yesterday
Pictures		Yesterday
Public		Yesterday
Templates		Yesterday
Videos		Yesterday
thinclient_drives		09:35

Step 2: Create a JavaScript file and execute it

2.1 Click on the **New File…** icon and create a JavaScript file named **FindMissingnumber.js**


Applications
FindMissingnumber.js - ...
Thu 20 Mar, 06:16 labuser
FindMissingnumber.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
FindMissingnur
New File...
Welcome
FindMissingnumber.js
FindMissingnumber.js
OUTLINE
TIMELINE
Ln 1, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

Note: You can name the JavaScript file according to your preference.

2.2 Add the code given below to the file created in step 1.3

```
function findMissingNumber(arr) {
const n = arr.length + 1; // Calculate n (since one number is missing in the array)
const expectedSum = (n * (n + 1)) / 2; // Sum of first n natural numbers
let actualSum = 0; // Initialize sum of array elements
for (let i = 0; i < arr.length; i++) {
actualSum += arr[i]; // Add each element to the sum
}

console.log(`Expected Sum: ${expectedSum}, Actual Sum: ${actualSum}`); // Debug output
return expectedSum - actualSum; // The missing number is the difference between expected and actual sums
}

// Example usage
const array = [1, 2, 4, 6, 3, 7, 8];
console.log(findMissingNumber(array)); // Output should be 5
```

.

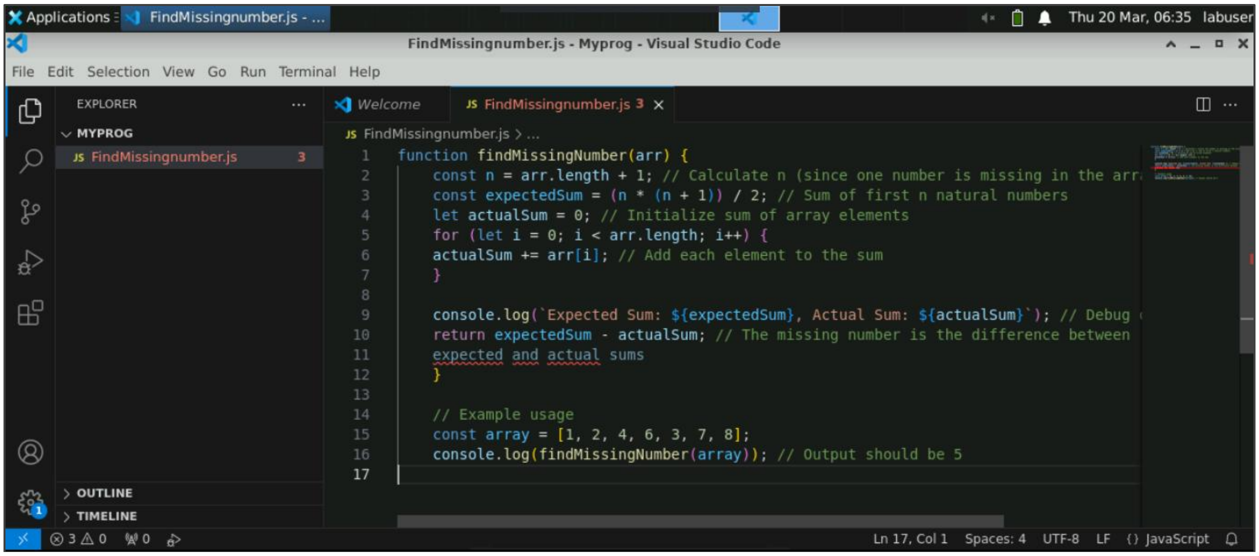
Note:

1. Calculate **n**, the total array length (including the missing number)
2. Use the formula **(n * (n + 1)) / 2** to find the sum of the initial **n** natural numbers
3. Sum all elements within the array

The difference between the expected sum (sum of the first **n** natural numbers) and the actual sum (sum of array elements) determines the missing number.

Assumptions:

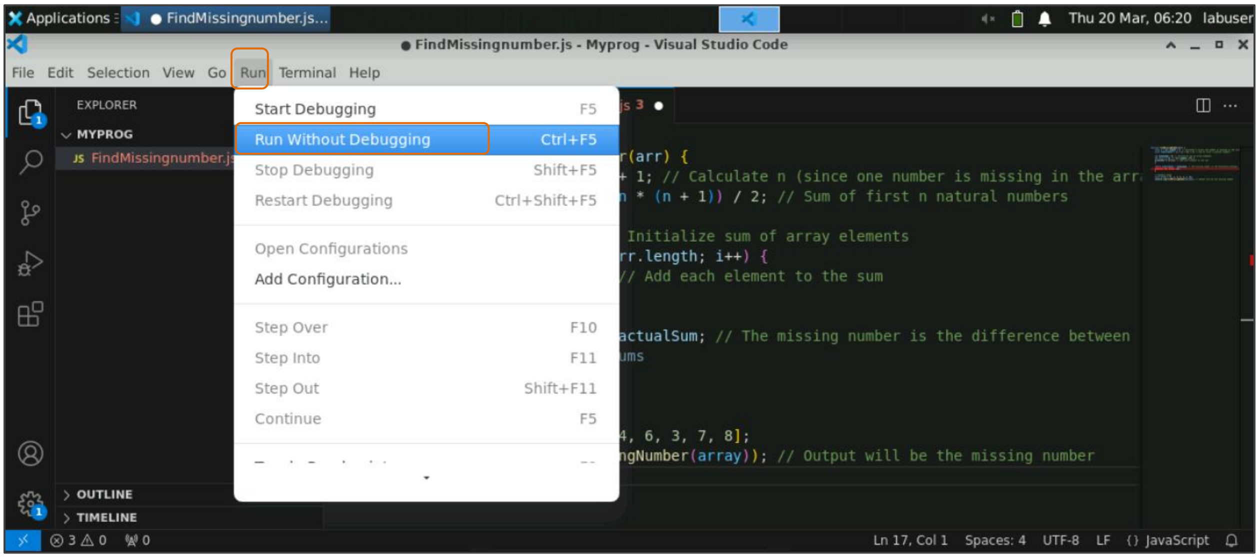
1. In this solution approach, we assume the absence of duplicate elements in the array.
2. This solution identifies only a single missing element.
3. Modifying the expected formula sum in the code allows finding missing numbers in various other arithmetic progression (AP) series.


Applications
FindMissingnumber.js - ...
Thu 20 Mar, 06:35 labuser
FindMissingnumber.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
FindMissingnumber.js 3
Welcome
FindMissingnumber.js 3
FindMissingnumber.js > ...
OUTLINE
TIMELINE
⊗ 3 ⚠ 0 0
Ln 17, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

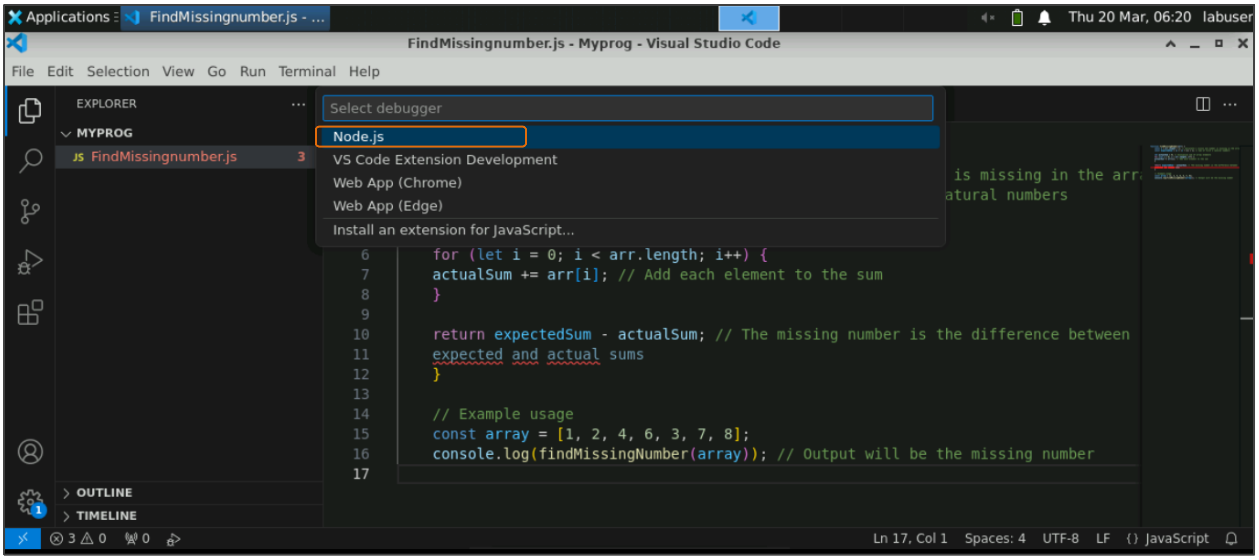
```
function findMissingNumber(arr) {
    const n = arr.length + 1; // Calculate n (since one number is missing in the arr
    const expectedSum = (n * (n + 1)) / 2; // Sum of first n natural numbers
    let actualSum = 0; // Initialize sum of array elements
    for (let i = 0; i < arr.length; i++) {
    actualSum += arr[i]; // Add each element to the sum
    }

    console.log(`Expected Sum: ${expectedSum}, Actual Sum: ${actualSum}`); // Debug
    return expectedSum - actualSum; // The missing number is the difference between
    expected and actual sums
    }

    // Example usage
    const array = [1, 2, 4, 6, 3, 7, 8];
    console.log(findMissingNumber(array)); // Output should be 5

```

.

2.3 Click **Run** and then **Run Without Debugging**. Select **Node.js** to check the output in the DEBUG CONSOLE.


Applications
FindMissingnumber.js...
Thu 20 Mar, 06:20 labuser
● FindMissingnumber.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
FindMissingnumber.js
Start Debugging F5
Run Without Debugging Ctrl+F5
Stop Debugging Shift+F5
Restart Debugging Ctrl+Shift+F5
Open Configurations
Add Configuration...
Step Over F10
Step Into F11
Step Out Shift+F11
Continue F5
OUTLINE
TIMELINE
Ln 17, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

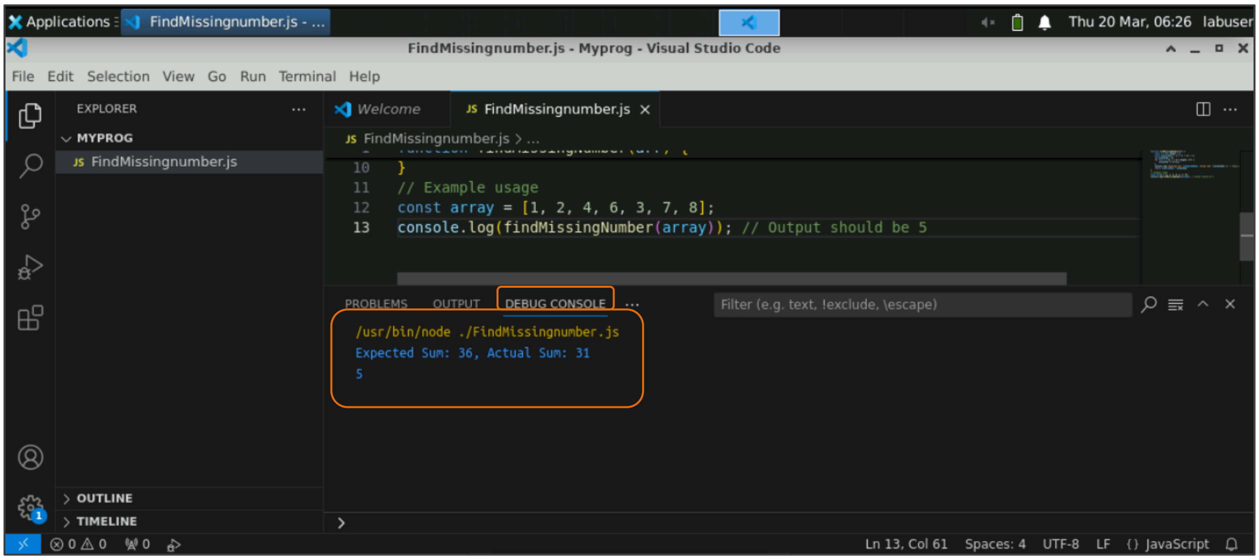
```
r(arr) {
+ 1; // Calculate n (since one number is missing in the arr
n * (n + 1)) / 2; // Sum of first n natural numbers

 Initialize sum of array elements
rr.length; i++) {
// Add each element to the sum

actualSum; // The missing number is the difference between
ums

4, 6, 3, 7, 8];
ngNumber(array)); // Output will be the missing number
```


Applications
FindMissingnumber.js - ...
Thu 20 Mar, 06:20 labuser
FindMissingnumber.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
FindMissingnumber.js 3
Select debugger
Node.js
VS Code Extension Development
Web App (Chrome)
Web App (Edge)
Install an extension for JavaScript...
OUTLINE
TIMELINE
Ln 17, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

```
 is missing in the arr
atural numbers
    for (let i = 0; i < arr.length; i++) {
    actualSum += arr[i]; // Add each element to the sum
    }

    return expectedSum - actualSum; // The missing number is the difference between
    expected and actual sums
    }

    // Example usage
    const array = [1, 2, 4, 6, 3, 7, 8];
    console.log(findMissingNumber(array)); // Output will be the missing number
```

.

2.4 View the output in the **DEBUG CONSOLE** as shown below:



```
}
// Example usage
const array = [1, 2, 4, 6, 3, 7, 8];
console.log(findMissingNumber(array)); // Output should be 5
```

```
/usr/bin/node ./FindMissingnumber.js
Expected Sum: 36, Actual Sum: 31
5
```

Applications
FindMissingnumber.js - ...
Thu 20 Mar, 06:26 labuser
FindMissingnumber.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
FindMissingnumber.js
Welcome
FindMissingnumber.js
FindMissingnumber.js > ...
PROBLEMS OUTPUT DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
OUTLINE
TIMELINE
Ln 13, Col 61 Spaces: 4 UTF-8 LF {} JavaScript

By following these steps, you have successfully created an algorithm for identifying a missing number in a sequence from 1 to n. This enhances logical thinking and foundational algorithm skills.