

1. Write a program Print Largest of three numbers



The image shows a code editor with a dark theme. The editor has a sidebar on the left with a file explorer showing a folder named 'Task' containing a file 'largestthree.js'. The main editor area displays the following JavaScript code:

```
Task] > largestthree.js
1  num1=20
2  num2=50
3  num3=30
4
5  if((num1>num2) && (num1>num3))
6  {
7      console.log(`largest is : ${num1}`);
8  }
9  else if(num2 > num3)
10 {
11     console.log(`largest number is: ${num2}`);
12 }
13 else
14 {
15     console.log(`largest number is: ${num3}`);
16 }
```

Below the code editor, there is a terminal window with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing the command 'node largestthree.js' being executed, which outputs 'largest number is: 50'.

```
PS C:\Users\HP\Desktop\MERN\06_JAVASCRIPT\Task] > node largestthree.js
largest number is: 50
PS C:\Users\HP\Desktop\MERN\06_JAVASCRIPT\Task] >
```

2.Difference Between While And Do while loop?

While Loop	Do-While Loop
In the While loop, the condition is tested before any statement is executed.	In Do while-loop, the statement is executed at least once even if the condition is false
Syntax: <code>while(condition){ // statements }</code>	Syntax: <code>do{ //statements }while(expression);</code>
In While loop, no semicolon is needed after the end of the condition.	In Do-while loop, semicolon needed after the end of the condition
While loop is an entry-controlled loop.	Do-while loop is an exit-controlled loop.
While loop may or may not be executed at all.	Do-while loop will execute at least once.
While loop can lead to errors if the condition is always false.	Do-while loop help prevents error as it runs at

2. Difference between break And Continue?

The Break Statement

You have already seen the break statement used in an earlier chapter of this tutorial. It was used to "jump out" of a switch() statement.

The break statement can also be used to jump out of a loop:

Example

```
for (let i = 0; i < 10; i++) {  
  if (i === 3) { break; }  
  text += "The number is " + i + "<br>";  
}
```

The Continue Statement

The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

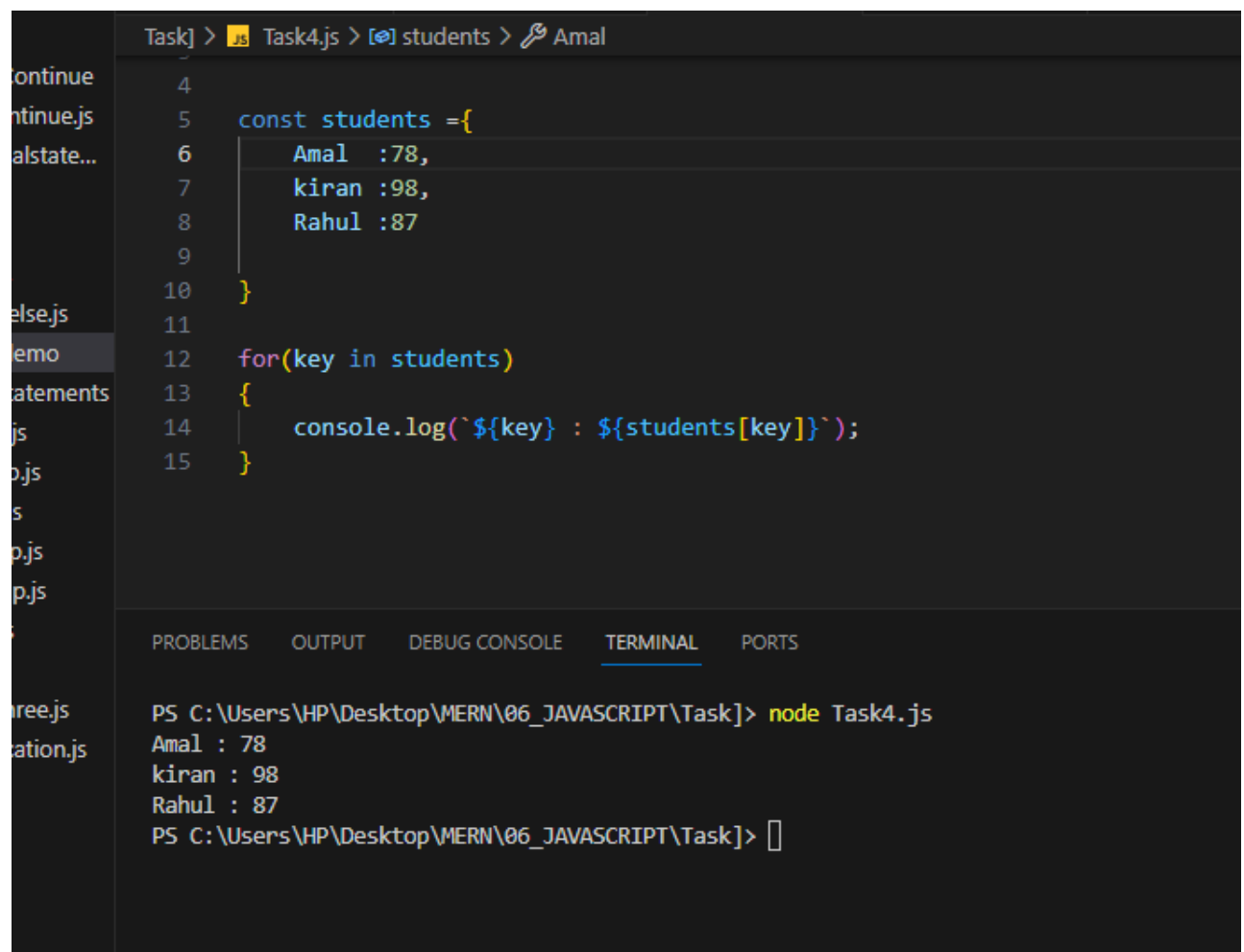
This example skips the value of 3:

Example

```
for (let i = 0; i < 10; i++) {  
  if (i === 3) { continue; }  
  text += "The number is " + i + "<br>";  
}
```

3. Define an Object with student names as properties and their marks as values?

Const students={amal 78 kiran 98 rahul 87}; using for in loop



```
Task] > .js Task4.js > [e] students > Amal
4
5  const students = {
6      Amal : 78,
7      kiran : 98,
8      Rahul : 87
9  }
10
11
12  for(key in students)
13  {
14      console.log(`${key} : ${students[key]}`);
15  }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\Desktop\MERN\06_JAVASCRIPT\Task] > node Task4.js
Amal : 78
kiran : 98
Rahul : 87
PS C:\Users\HP\Desktop\MERN\06_JAVASCRIPT\Task] > 
```