

JS BONUS ASSIGNMENT

Question 1 & 2:

1-

Js Sample Exam :

Question 1 :

1. What is an Event loop?

mechanism that handles the execution of various events or tasks within an app.

Commonly used in event-driven programming which make the processing of tasks more efficient.

2. To add an element at the beginning of an array?

Method 1 → `unshift()` `array.unshift(1);`

Method 2 → Spread Syntax : `newArray = [1, ...array];`

To add an element at the end:

Method 1 → `push()` `array.push(4);`

Method 2 → Spread Syntax : `newArray = [...array, 4]`

Question 2 : output for each

1. 3

2. 0 1 2 3 4

3. ['baz']

4. 1 Hello true

5. true

(2) [Array(2), Array(2)]

JS BONUS ASSIGNMENT

Question 3:

1-

```
JS Output.js x
C: > Users > 20114 > Desktop > JS Output.js > exampleObj > field2
1  function sumObjectValues(obj) {
2      let sum = 0;
3      for (let key in obj) {
4          if (obj.hasOwnProperty(key) && typeof obj[key] === 'number') {
5              sum += obj[key];
6          }
7      }
8      return sum;
9  }
10 const exampleObj = {
11     field1: 3,
12     field2: 5,
13     field3: 'not a number',
14     field4: 30,
15 };
16
17 const result = sumObjectValues(exampleObj);
18 console.log(result);
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

C:\Program Files\nodejs\node.exe .\Output.js

38

JS BONUS ASSIGNMENT

Question 3:

```
1- function sumObjectValues(obj) {  
    let sum = 0;  
    for (let key in obj) {  
        if (obj.hasOwnProperty(key) && typeof obj[key] === 'number') {  
            sum += obj[key];  
        }  
    }  
    return sum;  
}  
  
const exampleObj = {  
    field1: 3,  
    field2: 5,  
    field3: 'not a number',  
    field4: 30,  
};  
  
const result = sumObjectValues(exampleObj);  
console.log(result);
```

JS BONUS ASSIGNMENT

Question 3: 2-

```
JS Output.js ×
C: > Users > 20114 > Desktop > JS Output.js > ...
1  function asyncBlock1() {
2      console.log("Async Block 1: Start");
3      setTimeout(() => {
4          console.log("Async Block 1: Done");
5          asyncBlock2();
6      }, 2000);
7  }
8  function asyncBlock2() {
9      console.log("Async Block 2: Start");
10     setTimeout(() => {
11         console.log("Async Block 2: Done");
12         asyncBlock3();
13     }, 1500);
14 }
15 function asyncBlock3() {
16     console.log("Async Block 3: Start");
17     setTimeout(() => {
18         console.log("Async Block 3: Done");
19     }, 1000);
20 }
21
22 //sequence
23 asyncBlock1();
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Program Files\nodejs\node.exe .\Output.js
Async Block 1: Start
Async Block 1: Done
Async Block 2: Start
Async Block 2: Done
Async Block 3: Start
Async Block 3: Done
```

JS BONUS ASSIGNMENT

Question 3 :

2- `function asyncBlock1() {
 console.log("Async Block 1: Start");
 setTimeout(() => {
 console.log("Async Block 1: Done");
 asyncBlock2();
 }, 2000);
}
function asyncBlock2() {
 console.log("Async Block 2: Start");
 setTimeout(() => {
 console.log("Async Block 2: Done");
 asyncBlock3();
 }, 1500);
}
function asyncBlock3() {
 console.log("Async Block 3: Start");
 setTimeout(() => {
 console.log("Async Block 3: Done");
 }, 1000);
}

//sequence
asyncBlock1();`

JS BONUS ASSIGNMENT

Question 3: 3-

JS Output.js X

C: > Users > 20114 > Desktop > JS Output.js > ...

```
1  function getMaxValueAndIndex(numbersArray) {
2      if (!Array.isArray(numbersArray) || numbersArray.length === 0)
3      {
4          return null;
5      }
6      let maxIndex = 0;
7      let maxValue = numbersArray[0];
8      for (let i = 1; i < numbersArray.length; i++) {
9          if (numbersArray[i] > maxValue) {
10             maxValue = numbersArray[i];
11             maxIndex = i;
12         }
13     }
14     return { value: maxValue, index: maxIndex };
15 }
16 const numbers = [1,5,6,2,40,9];
17 const result = getMaxValueAndIndex(numbers);
18 if (result !== null) {
19     console.log(`Maximum value: ${result.value}`);
20     console.log(`Index of maximum value: ${result.index}`);
21 } else {
22     console.log("Invalid!!");
23 }
24
25
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

C:\Program Files\nodejs\node.exe .\Output.js

Maximum value: 40

Index of maximum value: 4

JS BONUS ASSIGNMENT

Question 3 :

3-

```
function getMaxValueAndIndex(numbersArray) {
  if (!Array.isArray(numbersArray) || numbersArray.length === 0)
  {
    return null;
  }
  let maxIndex = 0;
  let maxValue = numbersArray[0];
  for (let i = 1; i < numbersArray.length; i++) {
    if (numbersArray[i] > maxValue) {
      maxValue = numbersArray[i];
      maxIndex = i;
    }
  }
  return { value: maxValue, index: maxIndex };
}

const numbers = [1,5,6,2,40,9];
const result = getMaxValueAndIndex(numbers);
if (result !== null) {
  console.log(` Maximum value: ${result.value} `);
  console.log(` Index of maximum value: ${result.index} `);
} else {
  console.log("Invalid!!!");
}
```

JS BONUS ASSIGNMENT

Question 3: 4-

JS Output.js X

C: > Users > 20114 > Desktop > JS Output.js > ...

```
1  function dateDifferenceInDays(date1, date2) {
2      const oneDay = 24 * 60 * 60 * 1000;
3      const timeDiff = Math.abs(date2 - date1);
4      const daysDiff = Math.floor(timeDiff / oneDay);
5      return daysDiff;
6  }
7  // Ex
8  const startDate = new Date('2002-06-25');
9  const endDate = new Date('1998-07-3');
10
11  const days = dateDifferenceInDays(startDate, endDate);
12  console.log(`The difference between the dates is ${days} days.`);
13
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Program Files\nodejs\node.exe .\Output.js
The difference between the dates is 1453 days.
```


JS BONUS ASSIGNMENT

Question 3:

4-

```
function dateDifferenceInDays(date1, date2)
{
  const oneDay = 24 * 60 * 60 * 1000;

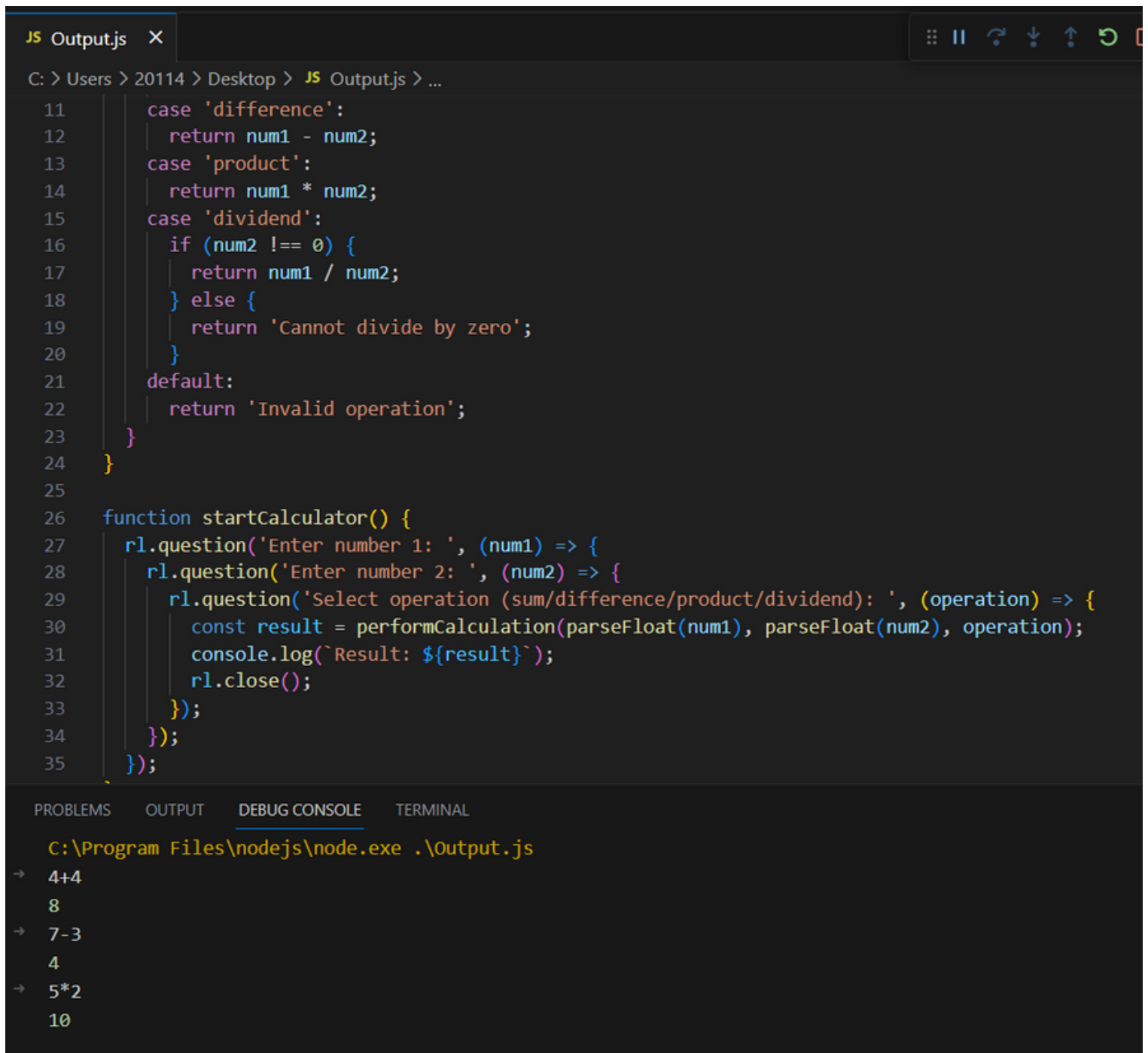
  const timeDiff = Math.abs(date2 - date1);
  const daysDiff = Math.floor(timeDiff / oneDay);

  return daysDiff;
}
// Ex
const startDate = new Date('2002-06-25');
const endDate = new Date('1998-07-3');

const days = dateDifferenceInDays(startDate, endDate);
console.log(` The difference between the dates is ${days} days.` );
```

JS BONUS ASSIGNMENT

Question 3: 5-



```
JS Output.js X
C: > Users > 20114 > Desktop > JS Output.js > ...

11     case 'difference':
12         return num1 - num2;
13     case 'product':
14         return num1 * num2;
15     case 'dividend':
16         if (num2 !== 0) {
17             return num1 / num2;
18         } else {
19             return 'Cannot divide by zero';
20         }
21     default:
22         return 'Invalid operation';
23 }
24 }
25
26 function startCalculator() {
27     rl.question('Enter number 1: ', (num1) => {
28         rl.question('Enter number 2: ', (num2) => {
29             rl.question('Select operation (sum/difference/product/dividend): ', (operation) => {
30                 const result = performCalculation(parseFloat(num1), parseFloat(num2), operation);
31                 console.log(`Result: ${result}`);
32                 rl.close();
33             });
34         });
35     });
36 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
C:\Program Files\nodejs\node.exe .\Output.js
→ 4+4
8
→ 7-3
4
→ 5*2
10
```

JS BONUS ASSIGNMENT

Question 3:

```
5- const readline = require('readline');
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

function performCalculation(num1, num2, operation) {
  switch (operation) {
    case 'sum':
      return num1 + num2;
    case 'difference':
      return num1 - num2;
    case 'product':
      return num1 * num2;
    case 'dividend':
      if (num2 !== 0) {
        return num1 / num2;
      } else {
        return 'Cannot divide by zero';
      }
    default:
      return 'Invalid operation';
  }
}

function startCalculator() {
  rl.question('Enter number 1:', (num1) => {
    rl.question('Enter number 2:', (num2) => {
      rl.question('Select operation (sum/difference/product/dividend):', (operation) => {
        const result = performCalculation(parseFloat(num1), parseFloat(num2), operation);
        console.log(`Result: ${result}`);
        rl.close();
      });
    });
  });
}

startCalculator();
```

JS BONUS ASSIGNMENT

Question 3: 6–

```
JS Output.js X
C: > Users > 20114 > Desktop > JS Output.js > getMultipleValues > value2
1  function getMultipleValues()
2  {
3      return {
4          value1: 25,
5          value2: 'HOLA',
6          value3: true
7      };
8  }
9
10 const result = getMultipleValues();
11 console.log(result.value1);
12 console.log(result.value2);
13 console.log(result.value3);
14
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Program Files\nodejs\node.exe .\Output.js
25
HOLA
true
```

JS BONUS ASSIGNMENT

Question 3:

6- `function getMultipleValues()`
`{`
 `return {`
 `value1: 25,`
 `value2: 'HOLA',`
 `value3: true`
 `};`
`}`

```
const result = getMultipleValues();  
console.log(result.value1);  
console.log(result.value2);  
console.log(result.value3);
```

JS BONUS ASSIGNMENT

Question 3: 7-

```
JS Output.js X
C: > Users > 20114 > Desktop > JS Output.js > ...
1  function reverseArray(arr) {
2      const reversed = [];
3      for (let i = arr.length - 1; i >= 0; i--) {
4          reversed.push(arr[i]);
5      }
6      return reversed;
7  }
8  const originalArray = [1,2,3,4,5,7,9];
9  const reversedArray = reverseArray(originalArray);
10 console.log(reversedArray);
11
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Program Files\nodejs\node.exe .\Output.js
> (7) [9, 7, 5, 4, 3, 2, 1]
```

JS BONUS ASSIGNMENT

Question 3:

7-

```
function reverseArray(arr) {  
  const reversed = [];  
  for (let i = arr.length - 1; i >= 0; i--) {  
    reversed.push(arr[i]);  
  }  
  return reversed;  
}  
  
const originalArray = [1,2,3,4,5,7,9];  
const reversedArray = reverseArray(originalArray);  
console.log(reversedArray);
```

JS BONUS ASSIGNMENT

Question 3:

8-

```
JS Output.js X
C: > Users > 20114 > Desktop > JS Output.js > ...
1  function objectToArray(obj)
2  {
3      const result = [];
4      for (const key in obj)
5      {
6          if (obj.hasOwnProperty(key))
7          {
8              result.push([key, obj[key]]);
9          }
10     }
11     return result;
12 }
13 const inputObject = { a: 1, b: 2 };
14 const outputArray = objectToArray(inputObject);
15 console.log(outputArray);
16

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
C:\Program Files\nodejs\node.exe .\Output.js
> (2) [Array(2), Array(2)]
```


JS BONUS ASSIGNMENT

Question 3:

8-

```
function objectToArray(obj)
{
  const result = [];
  for (const key in obj)
  {
    if (obj.hasOwnProperty(key))
    {
      result.push([key, obj[key]]);
    }
  }
  return result;
}

const inputObject = { a: 1, b: 2 };
const outputArray = objectToArray(inputObject);
console.log(outputArray);
```

JS BONUS ASSIGNMENT

Bonus ++

1-

JS Output.js X

C: > Users > 20114 > Desktop > JS Output.js > ...

```
1  function convertTo24HourFormat(time12Hour) {
2      const [time, period] = time12Hour.split(' ');
3      const [hours, minutes] = time.split(':');
4      let convertedHours = parseInt(hours);
5
6      if (period.toLowerCase() === 'pm' && convertedHours !== 12)
7      {
8          convertedHours += 12;
9      }
10     else if (period.toLowerCase() === 'am' && convertedHours === 12) {
11         convertedHours = 0;
12     }
13
14     const convertedTime = `${String(convertedHours).padStart(2, '0')}:${minutes}`;
15     return convertedTime;
16 }
17 const time12Hour = '04:30 PM';
18 const time24Hour = convertTo24HourFormat(time12Hour);
19 console.log(time24Hour);
20
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

C:\Program Files\nodejs\node.exe .\Output.js
16:30

JS BONUS ASSIGNMENT

Bonus ++

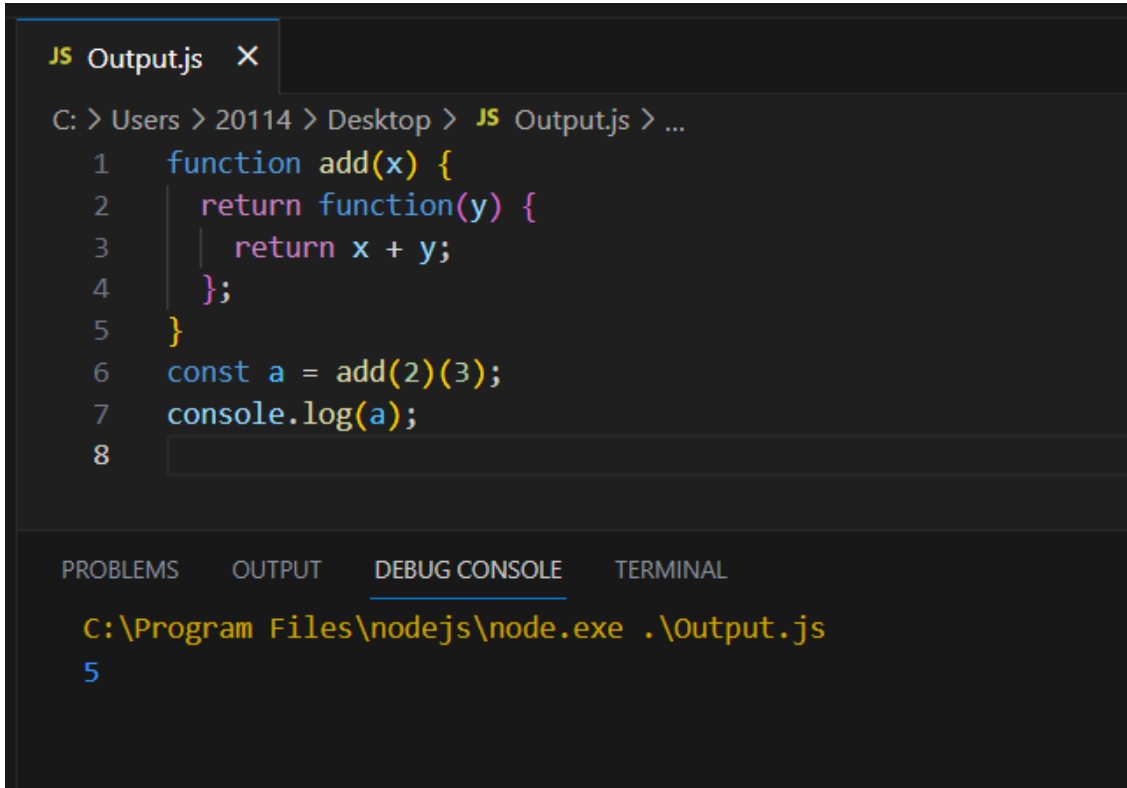
1-

```
function convertTo24HourFormat(time12Hour) {  
  const [time, period] = time12Hour.split(' ');  
  const [hours, minutes] = time.split(':');  
  let convertedHours = parseInt(hours);  
  if (period.toLowerCase() === 'pm' && convertedHours !== 12)  
  {  
    convertedHours += 12;  
  }  
  else if (period.toLowerCase() === 'am' && convertedHours === 12) {  
    convertedHours = 0;  
  }  
  
  const convertedTime = ` ${String(convertedHours).padStart(2, '0')}:${String(minutes).padStart(2, '0')} `;  
  return convertedTime;  
}  
  
const time12Hour = '04:30 PM';  
const time24Hour = convertTo24HourFormat(time12Hour);  
console.log(time24Hour);
```

JS BONUS ASSIGNMENT

Bonus ++

2-



```
JS Output.js ×
C: > Users > 20114 > Desktop > JS Output.js > ...
1  function add(x) {
2      return function(y) {
3          return x + y;
4      };
5  }
6  const a = add(2)(3);
7  console.log(a);
8

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
C:\Program Files\nodejs\node.exe .\Output.js
5
```

```
function add(x) {
  return function(y) {
    return x + y;
  };
}
const a = add(2)(3);
console.log(a);
```

JS BONUS ASSIGNMENT

Bonus ++

3-

JS Output.js X

C: > Users > 20114 > Desktop > JS Output.js > ...

```
1  const users = [  
2    { name: 'Mohamed', age: 45 },  
3    { name: 'John', age: 20 },  
4    { name: 'Ibrahiem', age: 28 }  
5  ];  
6  function doesUserExist(name, userList) {  
7    return userList.some(user => user.name === name);  
8  }  
9  const userNameToCheck = 'John';  
10 const userExists = doesUserExist(userNameToCheck, users);  
11 if (userExists) {  
12   console.log(`User ${userNameToCheck} exists.`);  
13 } else {  
14   console.log(`User ${userNameToCheck} does not exist.`);  
15 }  
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Program Files\nodejs\node.exe .\Output.js  
User John exists.
```

JS BONUS ASSIGNMENT

Bonus ++

```
3- const users = [
  { name: 'Mohamed', age: 45 },
  { name: 'John', age: 20 },
  { name: 'Ibrahiem', age: 28 }
];
function doesUserExist(name, userList) {
  return userList.some(user => user.name === name);
}
const userNameToCheck = 'John';
const userExists = doesUserExist(userNameToCheck, users);
if (userExists) {
  console.log(` User ${userNameToCheck} exists.` );
} else {
  console.log(` User ${userNameToCheck} does not exist.` );
}
```