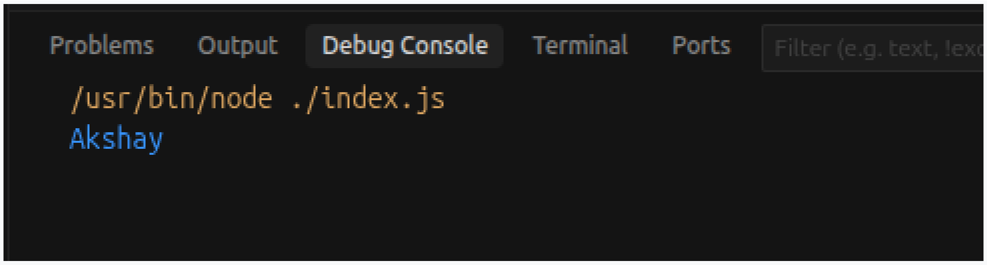
# Find and fix the bug from below code

**1**

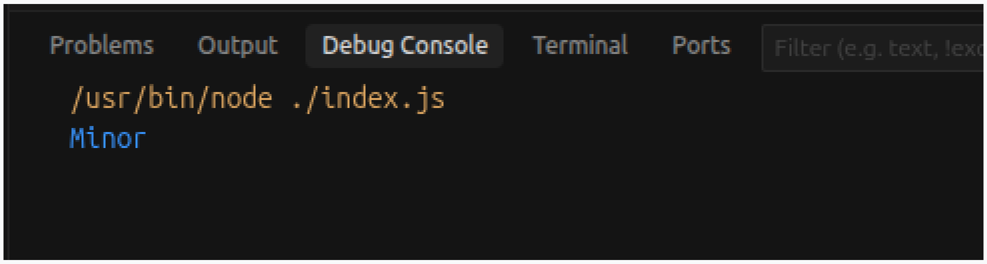
```
function printName() {
  if (true) {
    var name = "Akshay";
  }
  console.log(name);
 }
 printName();
```



```
/usr/bin/node ./index.js
Akshay
```

2

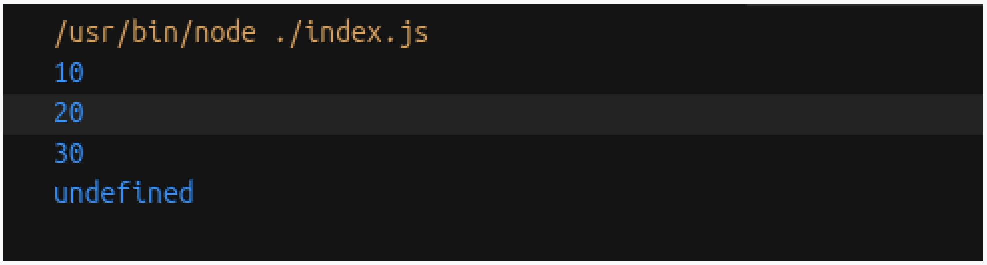
```
let age = "18";
if (age === 18) {

  console.log("Adult");

} else {

  console.log("Minor");

}
```



```
/usr/bin/node ./index.js
Minor
```

3.

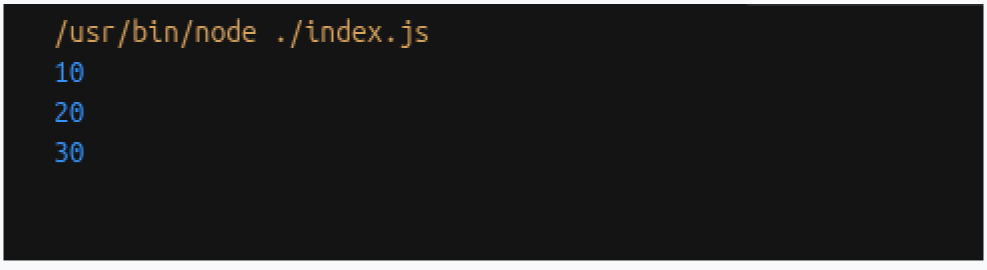
```
const arr = [10, 20, 30];

for (let i = 0; i <= arr.length; i++) {

  console.log(arr[i]);

}
```



```
/usr/bin/node ./index.js
10
20
30
undefined
```

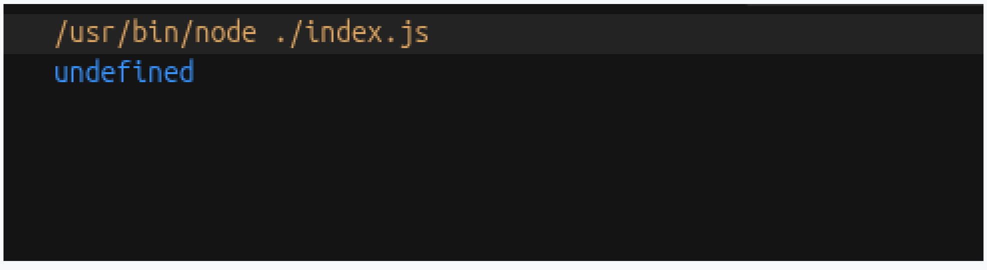
Explanation:
Here for loop is executing 4 times and array contains only 3 elements so it gives undefined
Below is corrected code:

```
const arr = [10, 20, 30];
for (let i = 0; i <arr.length; i++) {
  console.log(arr[i]);
}
```



```
/usr/bin/node ./index.js
10
20
30
```

```
4.let data;
setTimeout(() => {

  data = "Loaded";

}, 1000);

 console.log(data);
```

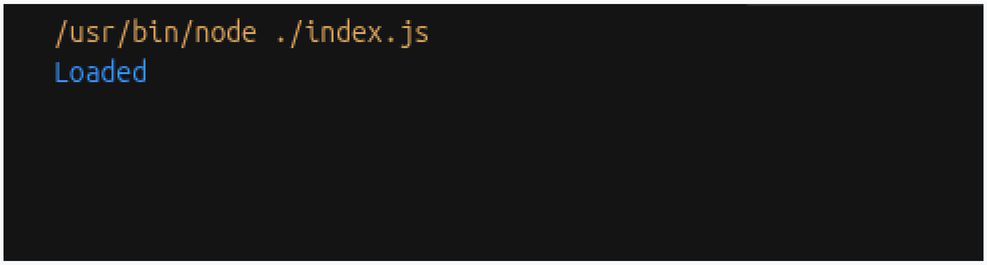


```
/usr/bin/node ./index.js
undefined
```

Explanation:
A variable data is created,but no value is assigned to it.
After that setTimeOut() is called for 1 sec. So JS code runs synchronously. So console.log() executes immediately. A variable data is empty so prints undefined.

Below is improved code

```
let data;
setTimeout(() => {
  data = "Loaded";
  console.log(data);
}, 1000);
```

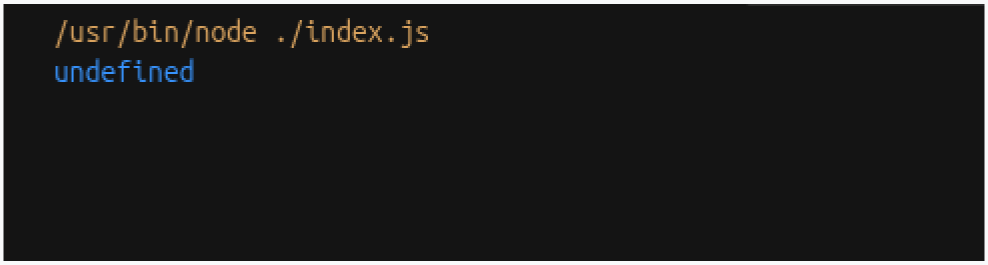


```
/usr/bin/node ./index.js
Loaded
```

Note:- loaded will print after 1 sec

5.

```
function add(a, b) {

  a + b;

}

const result = add(2, 3);

console.log(result);
```



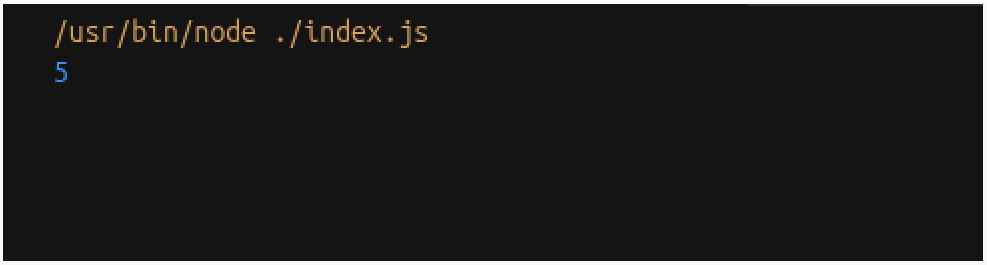
```
/usr/bin/node ./index.js
undefined
```

The function receives a = 2 and b = 3.
It calculates a + b, but does not return it.
Because nothing is returned, the function gives back undefined.
result becomes undefined.console.log(result) prints undefined.

Below is improved one:

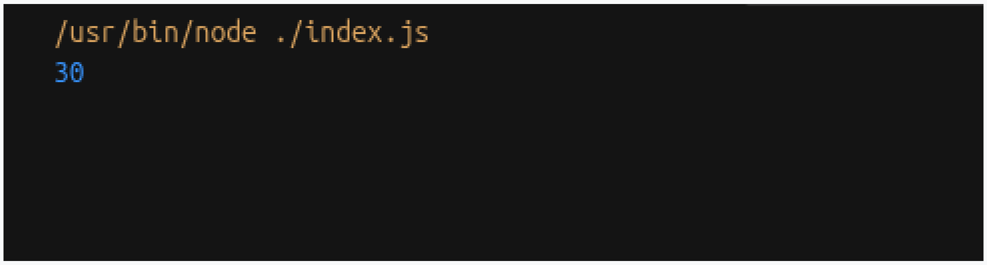
```
function add(a, b) {
  return a + b;
}
```

```
const result = add(2, 3);
console.log(result);
```



```
/usr/bin/node ./index.js
5
```

```
const user = {
  name: "John",
  age: 25,
};
function updateAge(u) {
  u.age = 30;
}
updateAge(user);
console.log(user.age);
```

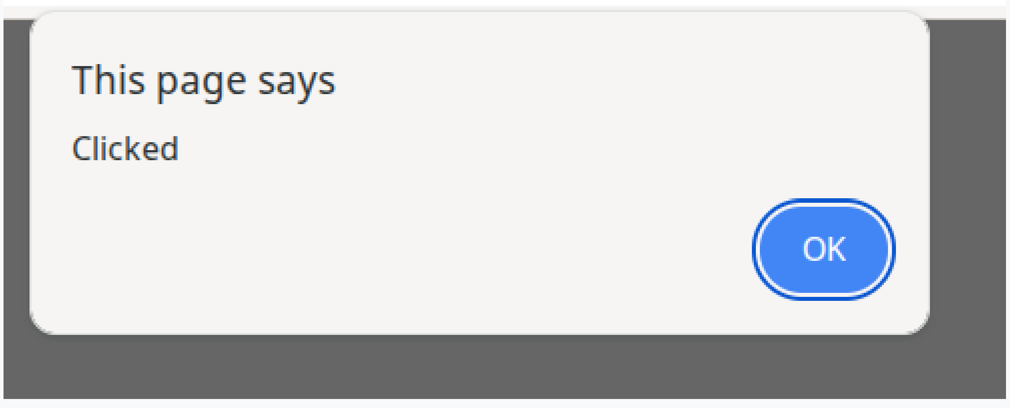


```
/usr/bin/node ./index.js
30
```

User is an object. In updateAge function the reference of an object is passed not a copy so original data gets affected. Age attribute is changed to 30

Note : If we want to pass copy of object we can used spread operator(...user)

```
<button id="btn">Click</button>

<script>

 const btn = document.getElementById("btn");

 btn.addEventListener("click", handleClick());

 function handleClick() {

   alert("Clicked");

 }</script>
```



This page says

Clicked

handleClick() is executed as soon as page loads. In addEventListener handleClick() function is passed not a result. It requires only the function name.

**Bug Fixed**

btn.addEventListener("click", handleClick);

Now function runs only after button is clicked

```
fetch("https://api.example.com/data")
 .then((res) => {
   res.json();
 })
 .then((data) => {
   console.log(data);
 });
```

fetch requests data from the server. The first then receives the response.

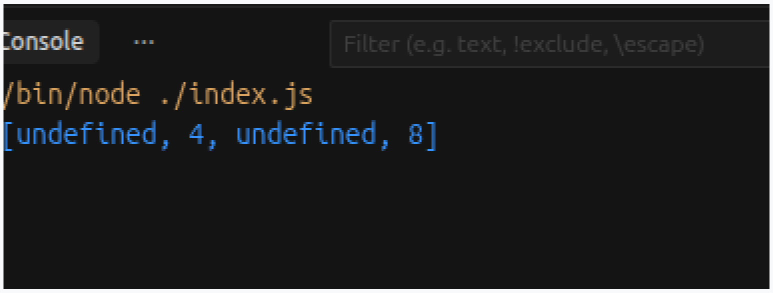
res.json() is called, but not returned. Because nothing is returned, the next then gets undefined. console.log(data) prints undefined.

Bug Fixed

```
fetch("https://api.example.com/data")
 .then((res) => res.json())
 .then((data) => {
   console.log(data);
 });
```

res.json() is returned The next step receives the parsed data So data contains the actual response

```
const nums = [1, 2, 3, 4];
const result = nums.map(□ => {
  if (n % 2 === 0) {
    return n * 2;
  }
});
console.log(result);
```



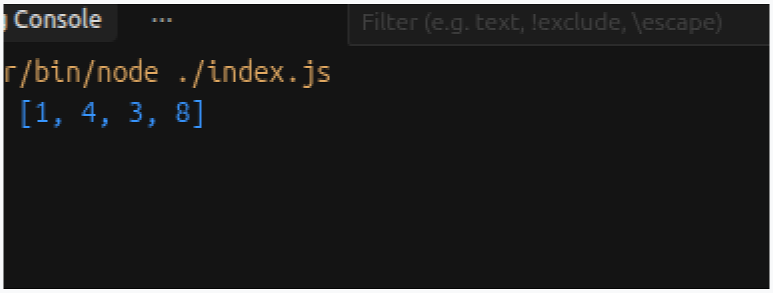
```
/bin/node ./index.js
[undefined, 4, undefined, 8]
```

map() function runs once for each element in nums array. For every even number it return its square and every odd number will be returned by undefined

Bug fixed :

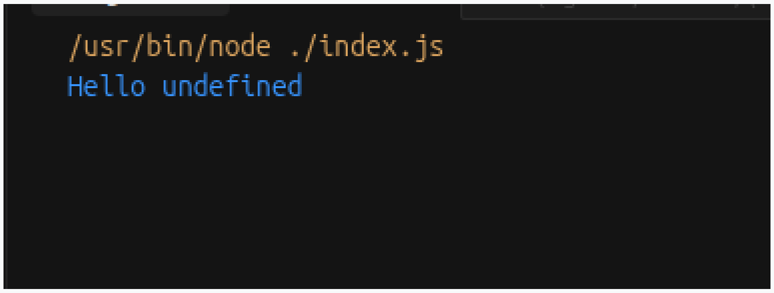
```
 const nums = [1, 2, 3, 4];
  const result = nums.map(n => {
    if (n % 2 === 0) {
      return n * 2;
    }
    return n
  });
```

```
console.log(result);
```



```
r/bin/node ./index.js
 [1, 4, 3, 8]
```

```
const person = {
  name: "Amar",
  greet: () => {

    console.log("Hello " +this.name);
  },

};

person.greet();
```



```
/usr/bin/node ./index.js
Hello undefined
```

greet is written using an arrow function. Arrow functions do not get their own this. this.name does not refer to person. So this.name is undefined. It prints: Hello undefined.

```
const person = {
  name: "Amar",
  greet() {
    console.log("Hello " + this.name);
  }
};
person.greet();
```