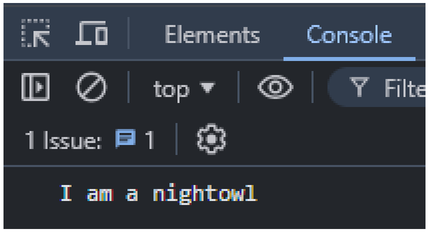
Closure:

TASK 1

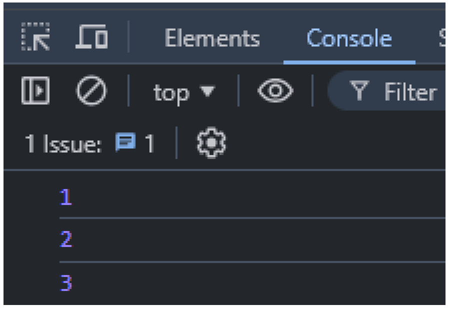
```
<script>function outer() {
  let loc = 'I am a nightowl';

  return function inner() {
    console.log(loc);
  };
}
const closure = outer();
closure();
</script>
```



```
I am a nightowl
```

TASK 2

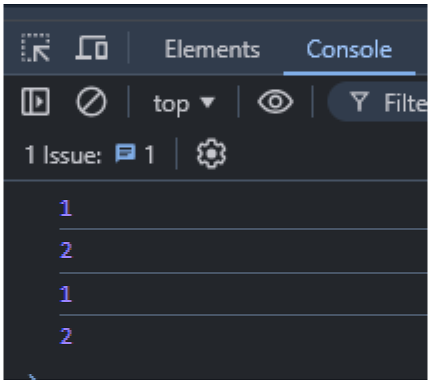
```
<script>
function counting(){
 let count = 0;
 return function(){
  count++;
  console.log(count);
 };
}

 const counter = counting()
  counter();
  counter();
  counter();


</script>
```



```
1
2
3
```

TASK 3

```
<script>
function counting(){
 let count = 0;
 return function(){
  count++;
  console.log(count);
 };
}

 const counter1 = counting();
 const counter2 = counting();
 counter1()
 counter1()


 counter2()
 counter2()

</script>
```

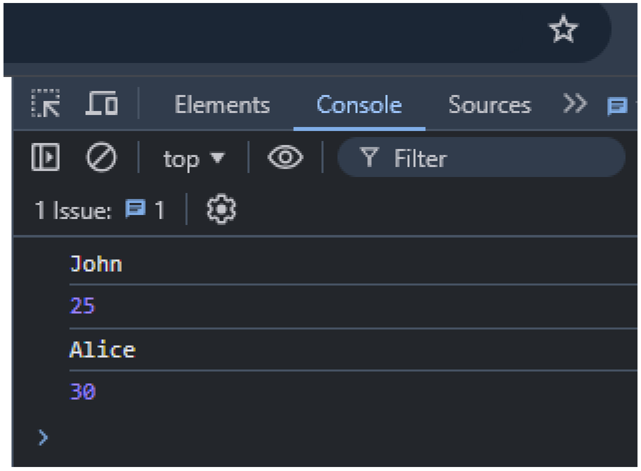


```
1
2
1
2
```

TASK 4

```
<script>
function createPerson(name, age) {
  let _name = name;
  let _age = age;

  return {
    getName: function() {
      return _name;
    },
    getAge: function() {
```

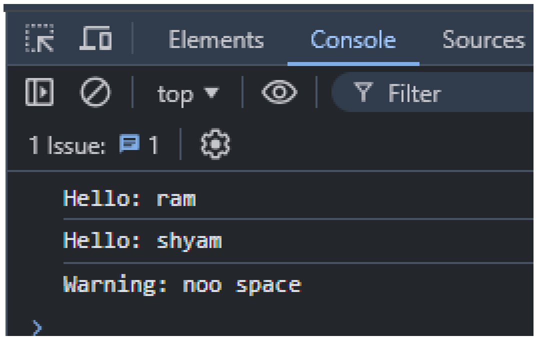
```
   return _age;
  },
  setName: function(newName) {
   _name = newName;
  },
  setAge: function(newAge) {
   _age = newAge;
  }
 };
}
const person = createPerson('John', 25);
console.log(person.getName());
console.log(person.getAge());
person.setName('Alice');
person.setAge(30);
console.log(person.getName());
console.log(person.getAge());
</script>
```



```
John
25
Alice
30
>
```

TASK 5

```
<script>
function functionFactory(prefix) {
 return function(message) {
  console.log(prefix + ': ' + message);
 };
}

const greet = functionFactory('Hello');
greet('ram');
greet('shyam');

const warn = functionFactory('Warning');
warn('noo space');

</script>
```



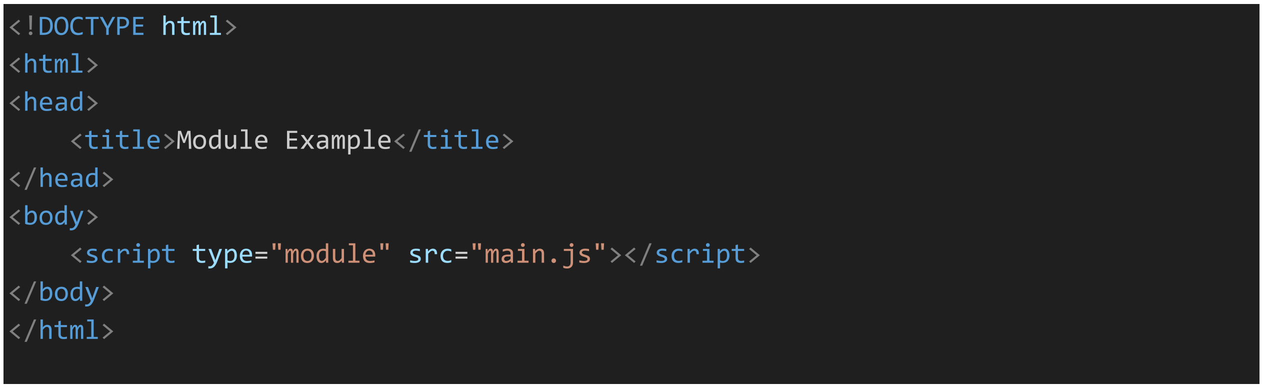
```
Hello: ram
Hello: shyam
Warning: noo space
```

6.MODULES,INTRODUCTION IMPORT AND EXPORT

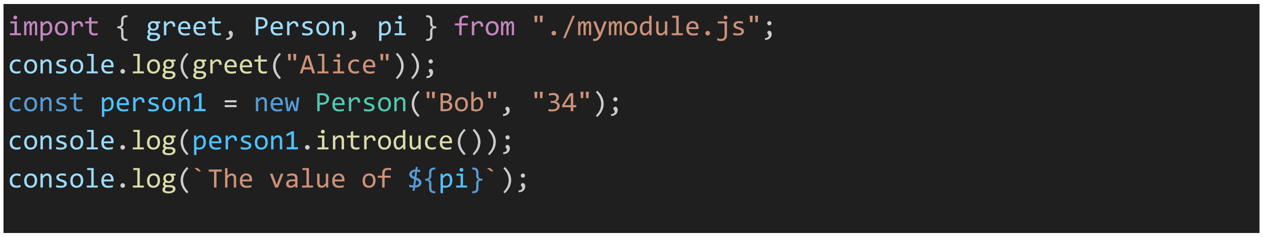
TASK-1

```
export function greet(name) {
  return `Hello ${name}`;
}
export class Person {
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }
  introduce() {
    return `Name: ${this.name}, Age: ${this.age}`;
  }
}
export const pi = 3.14;
```

TASK-2



```html
<!DOCTYPE html>
<html>
<head>
    <title>Module Example</title>
</head>
<body>
    <script type="module" src="main.js"></script>
</body>
</html>
```



```js
import { greet, Person, pi } from "./mymodule.js";
console.log(greet("Alice"));
const person1 = new Person("Bob", "34");
console.log(person1.introduce());
console.log(`The value of ${pi}`);
```

```
export function greet(name) {
    return `Hello ${name}`;
}
export class Person {
    constructor(name, age) {
        this.name = name;
        this.age = age;
    }
    introduce() {
        return `Name: ${this.name}, Age: ${this.age}`;
    }
}
export const pi = 3.14;
```

TASK 3

```
export function multiply(a, b) {
  return a * b;
}

export function subtract(a, b) {
  return a - b;
}

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

export class Person {
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }

  introduce() {
    return `My name is ${this.name}, Age is ${this.age}`;
  }
}

export const pi = 3.14;
```

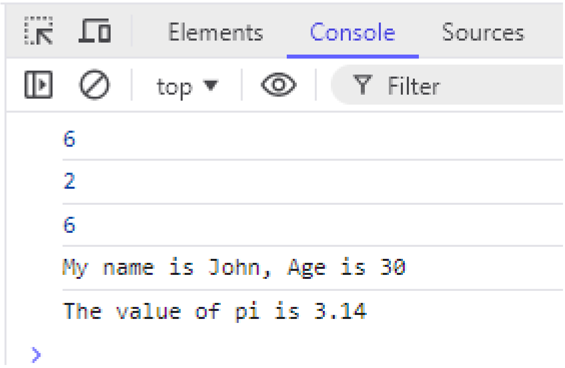
TASK 4

```
<!DOCTYPE html>
<html>
<head>
```

```html
    <title>Module Example</title>
</head>
<body>
    <script type="module" src="mymodule.js"></script>
</body>
</html>
```

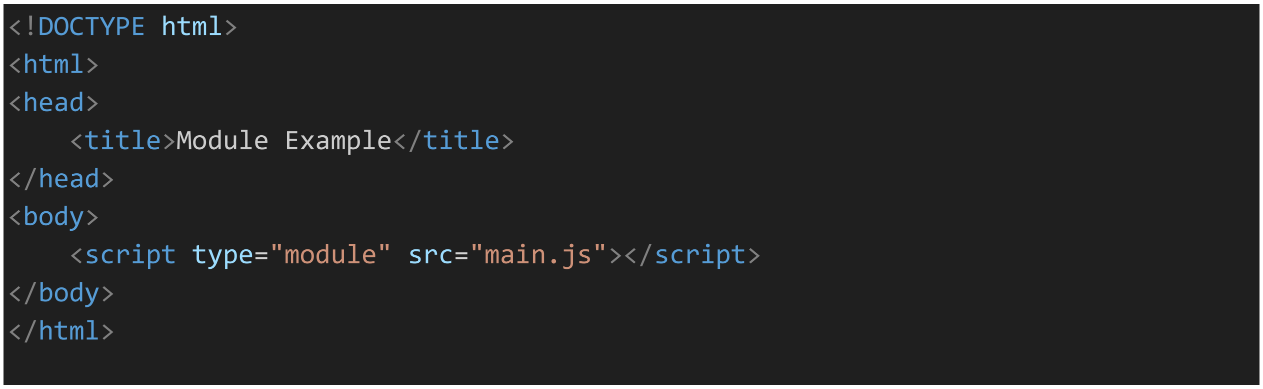
```javascript
export function multiply(a, b) {
    return a * b;
}
export function subtract(a, b) {
    return a - b;
}
export function add(a, b, c) {
    return a + b + c;
}
export class Person {
    constructor(name, age) {
        this.name = name;
        this.age = age;
    }
    introduce() {
        return `My name is ${this.name}, Age is ${this.age}`;
    }
}
export const pi = 3.14;
```

```javascript
import { multiply, subtract, add, Person, pi } from './main.js';

console.log(multiply(2, 3));
console.log(subtract(5, 3));
console.log(add(1, 2, 3));
const person1 = new Person('John', 30);
console.log(person1.introduce());
console.log(`The value of pi is ${pi}`);
```

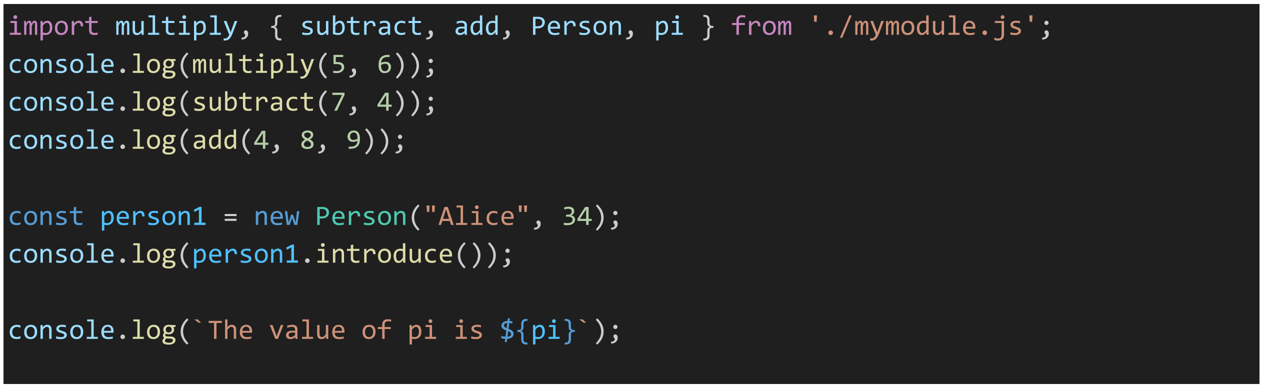


```
6
2
6
My name is John, Age is 30
The value of pi is 3.14
```

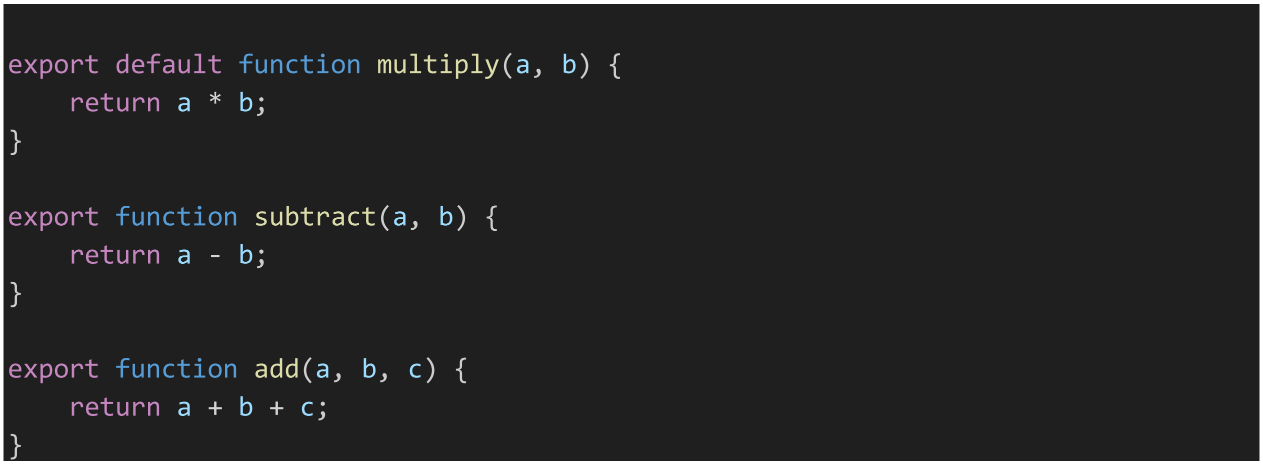
TASK 5



```html
<!DOCTYPE html>
<html>
<head>
    <title>Module Example</title>
</head>
<body>
    <script type="module" src="main.js"></script>
</body>
</html>
```

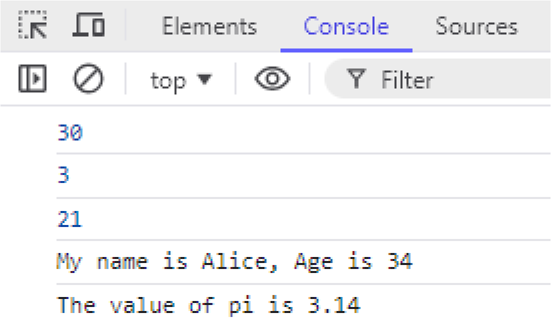


```javascript
import multiply, { subtract, add, Person, pi } from './mymodule.js';
console.log(multiply(5, 6));
console.log(subtract(7, 4));
console.log(add(4, 8, 9));

const person1 = new Person("Alice", 34);
console.log(person1.introduce());

console.log(`The value of pi is ${pi}`);
```



```javascript

export default function multiply(a, b) {
    return a * b;
}

export function subtract(a, b) {
    return a - b;
}

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

```
export class Person {
    constructor(name, age) {
        this.name = name;
        this.age = age;
    }

    introduce() {
        return `My name is ${this.name}, Age is ${this.age}`;
    }
}

export const pi = 3.14;
```



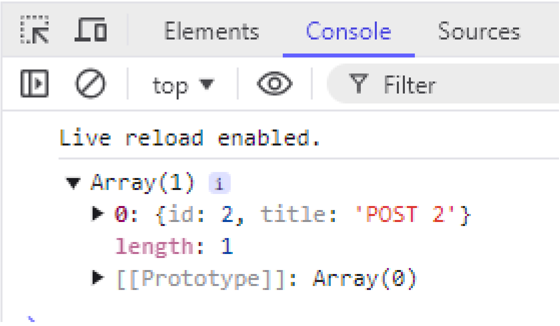
```
30
3
21
My name is Alice, Age is 34
The value of pi is 3.14
```

Async/await:

TASK 2

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,initial-scale=1.0" />
    <title>Task</title>
    <script>
        async function fetchAndProcessMockData() {
            const mockFetch = () =>
                new Promise((resolve) =>
                    setTimeout(() => resolve([{ id: 1, title: "Post 1" }, { id: 2, title: "Post 2" }]), 1000)
                );

            try {
                const data = await mockFetch();
                return data
```

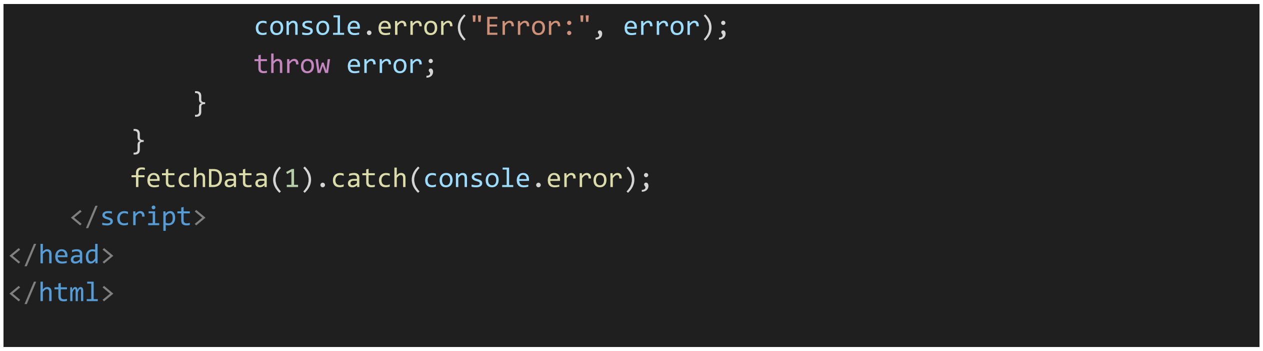
```
                    .filter((item) => item.id % 2 === 0)
                    .map((item) => ({ id: item.id, title:
item.title.toUpperCase() }));
            } catch (error) {
                console.error("Error:", error);
                throw error;
            }
        }

        fetchAndProcessMockData().then(console.log).catch(console.error);
    </script>
</head>
</html>
```



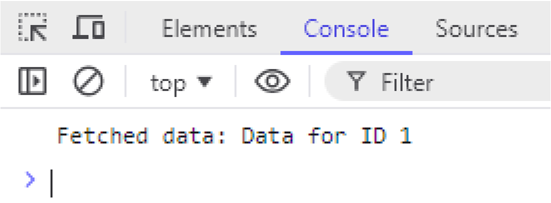
```
Live reload enabled.
▼ Array(1) i
  ▶ 0: {id: 2, title: 'POST 2'}
    length: 1
  ▶ [[Prototype]]: Array(0)
```

TASK 3

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,initial-scale=1.0" />
    <title>Task</title>
    <script>
        async function fetchData(id) {
            try {
                const mockFetch = (id) =>
                    new Promise((resolve, reject) => {
                        setTimeout(() => (id > 0 ? resolve(`Data for ID ${id}`) :
reject("Invalid ID")), 1000);
                    });
                const data = await mockFetch(id);
                console.log("Fetched data:", data);
                return data;
            } catch (error) {
```



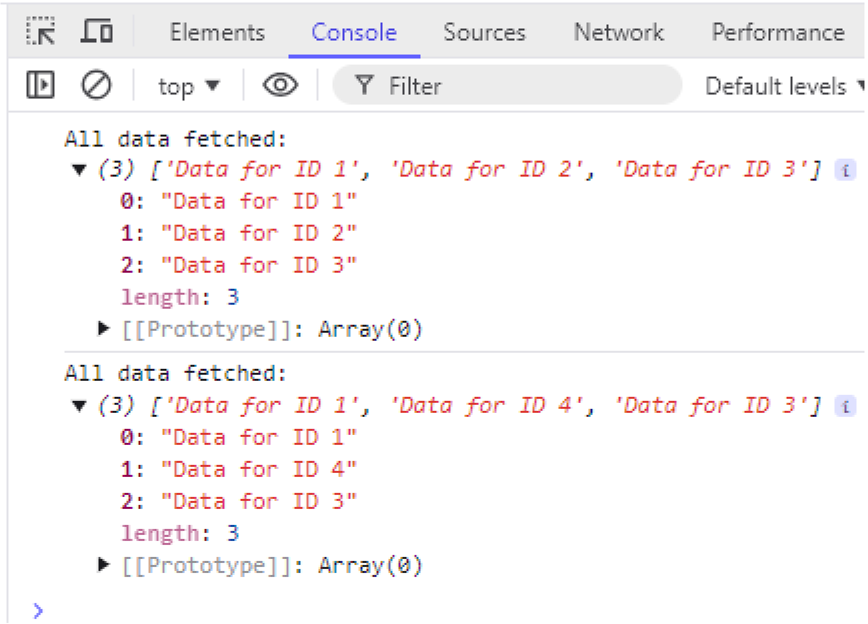
```
                console.error("Error:", error);
                throw error;
            }
        }
        fetchData(1).catch(console.error);
    </script>
</head>
</html>
```



```
Fetched data: Data for ID 1
>
```

TASK 4

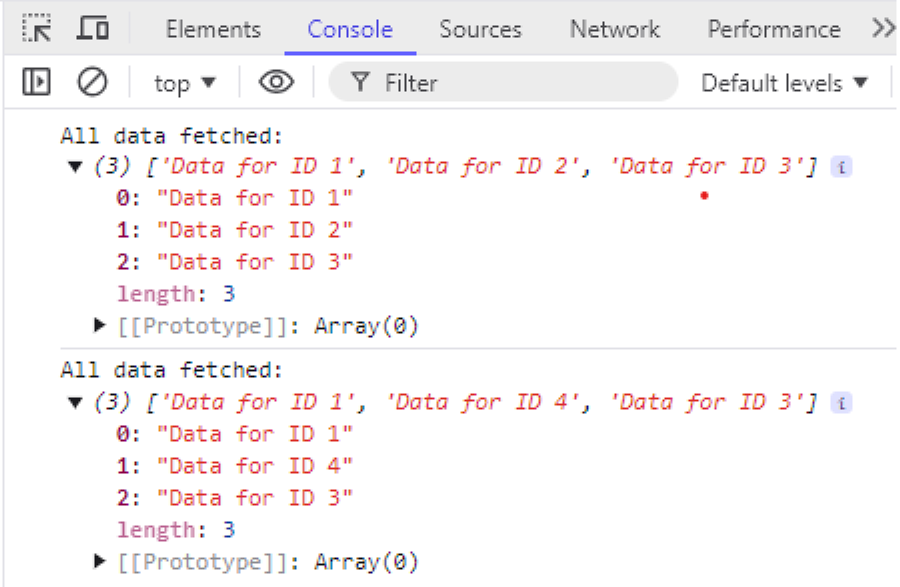
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1.0" />
  <title>Task</title>
  <script>
    async function fetchData(ids) {
      const mockFetch = (id) =>
        new Promise((resolve, reject) => {
          setTimeout(() => (id > 0 ? resolve(`Data for ID ${id}`) : reject("Invalid ID")), 1000);
        });

      try {
        const fetchPromises = ids.map(id => mockFetch(id));
        const results = await Promise.all(fetchPromises);
        console.log("All data fetched:", results);
        return results;
      } catch (error) {
        console.error("Error fetching data:", error);
      }
    }

    fetchData([1, 2, 3]).catch(console.error);
    fetchData([1, 4, 3]).catch(console.error);
  </script>
</head>
</html>
```



```
All data fetched:
▼ (3) ['Data for ID 1', 'Data for ID 2', 'Data for ID 3']
    0: "Data for ID 1"
    1: "Data for ID 2"
    2: "Data for ID 3"
    length: 3
  ▶ [[Prototype]]: Array(0)
All data fetched:
▼ (3) ['Data for ID 1', 'Data for ID 4', 'Data for ID 3']
    0: "Data for ID 1"
    1: "Data for ID 4"
    2: "Data for ID 3"
    length: 3
  ▶ [[Prototype]]: Array(0)
>
```

TASK 5

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1.0" />
  <title>Task</title>
  <script>
    async function processMultipleOperations() {
      const asyncOperation1 = () =>
        new Promise((resolve) => setTimeout(() => resolve("Operation 1 complete"), 1000));
      const asyncOperation2 = () =>
        new Promise((resolve) => setTimeout(() => resolve("Operation 2 complete"), 2000));
      const asyncOperation3 = () =>
        new Promise((resolve) => setTimeout(() => resolve("Operation 3 complete"), 1500));

      try {
        const results = await Promise.all([asyncOperation1(), asyncOperation2(), asyncOperation3()]);
        console.log("All operations completed:", results);
        return results;
      } catch (error) {
        console.error("Error in operations:", error);
      }
    }

    processMultipleOperations().then((results) => console.log(results));
  </script>
</head>
```



```
All data fetched:
▼ (3) ['Data for ID 1', 'Data for ID 2', 'Data for ID 3']
    0: "Data for ID 1"
    1: "Data for ID 2"
    2: "Data for ID 3"
    length: 3
  ▶ [[Prototype]]: Array(0)
All data fetched:
▼ (3) ['Data for ID 1', 'Data for ID 4', 'Data for ID 3']
    0: "Data for ID 1"
    1: "Data for ID 4"
    2: "Data for ID 3"
    length: 3
  ▶ [[Prototype]]: Array(0)
```