

PROMISES ASSIGNMENT

1).

```
function job() {  
    return new Promise(function(resolve, reject) {  
        reject();  
    });  
}  
  
let promise = job();  
promise  
    .then(function() {  
        console.log('Success 1');  
    })  
    .then(function() {  
        console.log('Success 2');  
    })  
    .then(function() {  
        console.log('Success 3');  
    })  
    .catch(function() {  
        console.log('Error 1');  
    })  
    .then(function() {  
        console.log('Success 4');  
    });
```

What is the output of the code above ?

- 2). Write a sleep function using a promise in javascript?
- 3). What is the output of the following code?

```
const promise = new Promise(res => res(2));
promise.then(v => {
    console.log(v);
    return v * 2;
})
    .then(v => {
        console.log(v);
        return v * 2;
    })
    .finally(v => {
        console.log(v);
        return v * 2;
    })
    .then(v => {
        console.log(v);
    });
```

4).

```
console.log('start')

const fn = () => (new Promise((resolve, reject) => {
    console.log(1);
    resolve('success')
})))

console.log('middle')

fn().then(res => {
    console.log(res)
})
```

```
console.log('end')
```

What is the output of this code snippet?

5). Write a function delay that returns a promise. And that promise should return a setTimeout that calls resolve after 1000ms.

6).

```
Promise.resolve('Success!')  
  .then(data => {  
    return data.toUpperCase()  
  })  
  .then(data => {  
    console.log(data)  
  })
```

What will the output be?

7).

```
var p = new Promise((resolve, reject) => {  
  reject(Error('The Fails!'))  
})  
  .catch(error => console.log(error))  
  .then(error => console.log(error))
```

What will the output be?

8).

```
console.log('start')

setTimeout(() => {
  console.log('setTimeout')
})

Promise.resolve().then(() => {
  console.log('resolve')
})

console.log('end')
```

What will the output be?

10).

```
console.log('start')

Promise.resolve(1).then((res) => {
  console.log(res)
})

Promise.resolve(2).then((res) => {
  console.log(res)
})

console.log('end')
```

What will the output be?