

Are you really listening? Let's see.

```
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 ?

- ☐ Error 1
- ☐ Success 1, Error 1
- ☐ Success 1, Success 2, Success 3, Success 4
- ☐ Success 1, Success 2, Success 3, Error 1, Success 4
- ☐ Error 1, Success 1, Success 2, Success 3, Success 4
- ☒ Error 1, Success 4

CHECK

```
// A
const add = (x, y, callback) => {
  setTimeout(() => {
    callback(x + y);
  }, 2);
};
```

```
// B
const add = (x, y, callback) => {
  setTimeout(() => {
    callback(x + y);
  }, 2000);
};
```

```
// C
const add = (x, y, callback) => {
  setTimeout(() => {
    callback(x + y, 2);
  });
};
```

```
// D
const add = (x, y, callback) => {
  setTimeout(() => {
    callback(x + y, 2000);
  });
};
```

Select one:

- ☐ A
- ☒ B ← CORRECT ANSWER
- ☐ C
- ☐ D
- ☐ None of the above

Correct!

```

function job(state) {
  return new Promise(function(resolve, reject) {
    if (state) {
      resolve('success');
    } else {
      reject('error');
    }
  });
}

let promise = job(true);

promise
  .then(function(data) {
    console.log(data);

    return job(true);
  })
  .then(function(data) {
    if (data !== 'victory') {
      throw 'Defeat';
    }

    return job(true);
  })
  .then(function(data) {
    console.log(data);
  })
  .catch(function(error) {
    console.log(error);

    return job(false);
  })
  .then(function(data) {
    console.log(data);

    return job(true);
  })
  .catch(function(error) {
    console.log(error);

    return 'Error caught';
  })
  .then(function(data) {
    console.log(data);

    return new Error('test');
  })
  .then(function(data) {
    console.log('Success:', data.message);
  })
  .catch(function(data) {
    console.log('Error:', data.message);
  });

```

✔ What is the output of the code above?

- ☐ error, error, Error caught, Error: test
- ☐ success, success, Error caught, Success: Test
- ☒ success, Defeat, error, Error caught, Success: test
- ☐ error, Error caught, Success: test
- ☐ success, Defeat, error, Error caught, Error: test
- ☐ success, error, Defeat, Success: test

CHECK

```
function job(state) {
  return new Promise(function(resolve, reject) {
    if (state) {
      resolve('success');
    } else {
      reject('error');
    }
  });
}

let promise = job(true);

promise
  .then(function(data) {
    console.log(data);

    return job(false);
  })
  .catch(function(error) {
    console.log(error);

    return 'Error caught';
  })
  .then(function(data) {
    console.log(data);

    return job(true);
  })
  .catch(function(error) {
    console.log(error);
  });
```

✓ What is the output of the code above?

- ☐ error, success, Error caught
- ☐ success, success
- ☐ success, error, success, error
- ☒ success, error, Error caught
- ☐ error, Error caught, success
- ☐ error, Error caught, success, error
- ☐ success, error, error
- ☐ success, success, success

CHECK