Challenge 8

```
1)const sleep = (ms) => {
  return new Promise((resolve) => {
    setTimeout(resolve, ms);
  });
};
console.log("Start");
sleep(2000)
  .then(() => {
    console.log("After 2 seconds");
    return sleep(1000);
  })
  .then(() => {
    console.log("After 1 more second");
  })
  .then(() => {
    console.log("End");
  });
```

```
2)function delay(ms) {
  return new Promise(resolve => {
    setTimeout(() => {
      resolve(`Promise resolved after ${ms} milliseconds`);
    }, ms);
  });
}
delay(2000)
  .then(result => {
    console.log(result);
  })
  .catch(error => {
    console.error(error);
  });

3)function randomResolveReject() {
  return new Promise((resolve, reject) => {
    const random = Math.random();
    if (random >= 0.5) {
      resolve(`Resolved with random number: ${random}`);
    } else {
      reject(`Rejected with random number: ${random}`);
    }
  });
}
randomResolveReject()
  .then(result => {
    console.log(result);
  })
  .catch(error => {
    console.error(error);
  });
```

```
4)function allPromises(promises) {
  return new Promise((resolve, reject) => {
    Promise.all(promises)
      .then(results => {
        resolve(results);
      })
      .catch(error => {
        reject(error);
      });
  });
}
const promise1 = Promise.resolve('Promise 1 resolved');
const promise2 = Promise.reject('Promise 2 rejected');
const promise3 = Promise.resolve('Promise 3 resolved');
const promiseArray = [promise1, promise2, promise3];
allPromises(promiseArray)
  .then(results => {
    console.log('All Promises resolved:', results);
  })
  .catch(error => {
    console.error('Promise rejection:', error);
  });
```