1. 下面代码的输出结果是什么

```
const promise = new Promise((resolve, reject) => {
    console.log(1);
    resolve();
    console.log(2);
})

promise.then(() => {
    console.log(3);
})

console.log(4);
```

2. 下面代码的输出结果是什么

```
const promise = new Promise((resolve, reject) => {
    console.log(1);
    setTimeout(()=>{
      console.log(2)
      resolve();
        console.log(3);
    })
})

promise.then(() => {
    console.log(4);
})

console.log(5);
```

3. 下面代码的输出结果是什么

```
const promise1 = new Promise((resolve, reject) => {
        setTimeout(() => {
      resolve()
  }, 1000)
})
const promise2 = promise1.catch(() => {
  return 2;
})

console.log('promise1', promise1)
console.log('promise2', promise2)

setTimeout(() => {
  console.log('promise1', promise1)
  console.log('promise2', promise2)
}, 2000)
```

4. 下面代码的输出结果是什么

```
async function m(){
  const n = await 1;
  console.log(n);
}

m();
console.log(2);
```

5. 下面代码的输出结果是什么

```
async function m(){
  const n = await 1;
  console.log(n);
}

(async ()=>{
  await m();
  console.log(2);
})();

console.log(3);
```

6. 下面代码的输出结果是什么

```
async function m1(){
  return 1;
}

async function m2(){
  const n = await m1();
  console.log(n)
  return 2;
}

async function m3(){
  const n = m2();
  console.log(n);
  return 3;
}

m3().then(n=>{
  console.log(n);
});

m3();

console.log(4);
```

7. 下面代码的输出结果是什么

```
Promise.resolve(1)
  .then(2)
  .then(Promise.resolve(3))
  .then(console.log)
```

8. 下面代码的输出结果是什么

```
var a;
var b = new Promise((resolve, reject) => {
  console.log('promise1');
  setTimeout(()=>{
    resolve();
  }, 1000);
}).then(() => {
  console.log('promise2');
}).then(() => {
  console.log('promise3');
}).then(() => {
  console.log('promise4');
});

a = new Promise(async (resolve, reject) => {
  console.log(a);
  await b;
  console.log(a);
  console.log('after1');
  await a;
  resolve(true);
  console.log('after2');
});

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

9. 下面代码的输出结果是什么

```
async function async1() {
    console.log('async1 start');
    await async2();
    console.log('async1 end');
}
async function async2() {
        console.log('async2');
}

console.log('script start');

setTimeout(function() {
    console.log('setTimeout');
}, 0)

async1();

new Promise(function(resolve) {
    console.log('promise1');
    resolve();
}).then(function() {
    console.log('promise2');
});
console.log('script end');
```