Question 1

Not yet answered

Marked out of 1.00

What will be the output of the following code?

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

setTimeout(() => console.log('timeout 1'), 0);

Promise.resolve().then(() => {
  console.log('promise 1');
  setTimeout(() => console.log('timeout 2'), 0);
});

queueMicrotask(() => console.log('microtask'));

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

Select one:

○ a. start microtask end promise 1 timeout 1 timeout 2

○ b. start end promise 1 microtask timeout 1 timeout 2

○ c. start end timeout 1 promise 1 timeout 2 microtask

○ d. start end microtask promise 1 timeout 1 timeout 2

Question 2

Not yet answered

Marked out of 1.00

What is logged to the console?

```
function counter() {
  let count = 0;
  return {
    inc: () => ++count,
    dec: () => --count
  };
}

const a = counter();
const b = counter();

a.inc(); a.inc();
b.inc();

console.log(a.dec(), b.dec());
```

Select one:

○ a. 2 0

○ b. 1 1

○ c. 2 1

○ d. 1 0

What will be printed?

```
function A() {}
A.prototype.value = 1;

const a1 = new A();
A.prototype = { value: 2 };

const a2 = new A();

console.log(a1.value, a2.value);
```

Select one:

○ a. 1 2

○ b. undefined 2

○ c. 2 2

○ d. 1 1

What will the following code print?

```
let obj = {
  valueOf() {
    return {};
  },
  toString() {
    return "5";
  }
};

console.log(obj == 5);
console.log(obj === 5);
```

Select one:

○ a. true true

○ b. false true

○ c. true false

○ d. false false

What is the output of this code?

```
async function* gen() {
  yield await Promise.resolve(1);
  yield 2;
}

(async () => {
  for await (const val of gen()) {
    console.log(val);
  }
})();
```

Select one:

- a. undefined 2
- b. 1 undefined
- c. Promise { 1 } 2
- d. 1 2