

1. HTTP Server

Snippet 1:

```
const { createServer } = require('http');
const { parse } = require('url');
const { readFile } = require('fs').promises;

const server = createServer(async (req, res) => {
  const { pathname } = parse(req.url);

  try {
    const data = await readFile(`.${pathname}`, 'utf8');
    res.writeHead(200, { 'Content-Type': 'text/plain' });
    res.end(data);
  } catch (error) {
    res.writeHead(404, { 'Content-Type': 'text/plain' });
    res.end('File not found');
  }
});

server.listen(3000);
```

Snippet 2:

```
const { createServer } = require('http');
const { parse } = require('url');
const { readFile } = require('fs').promises;

const server = createServer((req, res) => {
  const { pathname } = parse(req.url);

  readFile(`.${pathname}`, 'utf8')
    .then((data) => {
      res.writeHead(200, { 'Content-Type': 'text/plain' });
      res.end(data);
    })
    .catch((error) => {
      res.writeHead(404, { 'Content-Type': 'text/plain' });
      res.end('File not found');
    });
});

server.listen(3000);
```

Which of the following statements are true?

Pick **ONE OR MORE** options

- ☐ Both implementations will work as expected, but the error handling in Snippet 1 is better due to the use of `async/await`.
- ☐ If the request URL pathname is `/`, both Snippet 1 and Snippet 2 will throw an error, as they attempt to read the file `./`.
- ☐ If the request URL pathname is a directory instead of a file, both implementations will throw an error and return a 404 status code.
- ☐ Both implementations will return a 404 status code if the requested file has an encoding other than 'utf8.'

[Clear Selection](#)

2. String and ASCII Values

```
const fetchData = () => {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve([
        { id: 1, value: "A" },
        { id: 2, value: "B" },
        { id: 3, value: "C" },
      ]);
    }, 2000);
  });
};

const processData = async (transform) => {
  const data = await fetchData();
  return data.map((item) => transform(item));
};

const transformData = (item) => {
  const transformedValue = String.fromCharCode(
    item.value.charCodeAt(0) + item.id
  );
  return { ...item, value: transformedValue };
};

(async () => {
  const transformedData = await processData(transformData);
  console.log(transformedData);
})();
```

What is the output of the `console.log(transformedData)` statement?

Pick **ONE** option

- ☐ [{ id: 1, value: "B" }, { id: 2, value: "D" }, { id: 3, value: "F" }]
- ☐ [{ id: 1, value: "C" }, { id: 2, value: "D" }, { id: 3, value: "F" }]
- ☐ [{ id: 1, value: "C" }, { id: 2, value: "E" }, { id: 3, value: "F" }]
- ☐ [{ id: 1, value: "C" }, { id: 2, value: "E" }, { id: 3, value: "G" }]

[Clear Selection](#)