



```
20 app.listen(port, () => {
21   console.log(`server đang chạy trên cổng ${port}`);
22 }
23
24
25
26 setTimeout(() => {
27   console.log("Đã hết 5 giây! (sau 5 giây)");
28 },
29 5000);
30
31 console.log("Đang chờ");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22631.5039]  
(c) Microsoft Corporation. All rights reserved.

D:\Phát triển ứng dụng web\teoshop>node index.js  
Debugger attached.  
Đang chờ  
server đang chạy trên cổng 9000  
Đã hết 5 giây! (sau 5 giây)

```

25
26 function
27 ✓ getUser(userId, callback) {
28
29     fetch(`https://jsonplaceholder.typicode.com/users/${userId}`)
30
31     .then(response => response.json())
32
33     .then(data => callback(null, data)) // Thành công, gọi callback với dữ liệu
34
35     .catch(error => callback(error, null)); // Lỗi, gọi callback với lỗi
36
37 }
38
39
40
41 ✓ getUser(1, function(error, user) {
42
43     ✓ if (error) {
44
45         console.error("Lỗi:", error);
46
47     }
48     ✓ else {
49
50         console.log("Tên người dùng:", user.name);
51
52     }
53
54 });
55

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

Xử lý xong!
Tiếp tục công việc khác
1
4
9
16
Tên người dùng: Leanne Graham
Waiting for the debugger to disconnect...

```

Câu 10.2 Hàm callback trong JavaScript là gì? Phát biểu nào sau đây không đúng?

- A. Hàm callback là một hàm được truyền vào một hàm khác như một tham số.
- B. Hàm callback thường được sử dụng để xử lý các tác vụ bất đồng bộ.
- C. Hàm callback có thể là hàm ẩn danh, hàm mũi tên hoặc hàm đã được định nghĩa trước.
- D. Hàm callback được thực thi ngay lập tức sau khi hàm "cha" bắt đầu thực hiện.**

Hàm callback không được thực thi ngay lập tức mà nó được thực thi sau khi hàm "cha" thực hiện xong một tác vụ nào đó, đặc biệt là các tác vụ bất đồng bộ.

Câu 10.3 Callback hell trong JavaScript là gì? Phát biểu nào sau đây không đúng?

A. Callback hell xảy ra khi có quá nhiều hàm callback lồng nhau, khiến mã nguồn khó đọc và bảo trì.

B. Callback hell thường xuất hiện khi xử lý các tác vụ bất đồng bộ phức tạp.

C. Callback hell là một cách hiệu quả để quản lý các tác vụ bất đồng bộ trong JavaScript.

Callback hell gây khó khăn trong việc quản lý và bảo trì mã nguồn, nó không phải là một cách hiệu quả.

D. Callback hell có thể được giải quyết bằng Promise hoặc async/await.

- Hàm callback là một hàm được truyền vào một hàm khác như một tham số và được thực thi sau khi hàm "cha" hoàn thành một tác vụ.
- Callback hell là tình trạng mã nguồn trở nên khó đọc và bảo trì do quá nhiều hàm callback lồng nhau.