

Question 1: You have an asynchronous function that fetches data from an external API using `axios`. Sometimes, the API is slow to respond or fails altogether. You need to implement a solution that retries the request up to 3 times if it fails, but with an increasing delay between retries (e.g., 1 second, 2 seconds, 4 seconds). Write a Node.js function to achieve this behavior.

Question 2: Write a JavaScript function that takes an array of integers and returns a new array where each element at index `i` of the new array is the product of all the numbers in the original array except the one at `i`.

For example, given `[1, 2, 3, 4]`, the function should return `[24, 12, 8, 6]`.

Bonus: Can you solve it without using division?

Question 3:

Consider the following schema for a simple e-commerce application:

- **customers table:** Contains customer details.
 - `customer_id` (Primary Key)
 - `customer_name`
- **orders table:** Contains order details for customers.
 - `order_id` (Primary Key)
 - `customer_id` (Foreign Key referencing `customers.customer_id`)
 - `order_date`
- **order_items table:** Contains the items in each order.
 - `item_id` (Primary Key)
 - `order_id` (Foreign Key referencing `orders.order_id`)
 - `product_name`
 - `quantity`

Task: Write a SQL query to delete a customer and all their related orders and order items. You should ensure that all related data is deleted from the `orders` and `order_items` tables when a customer is deleted.

Question 4: What will be the output of the following code? Explain why.

```
const arr = [1, 2, 3];  
  
const newArr = arr.map(x => { arr.push(x + 3); return x; });  
  
console.log(newArr);  
  
console.log(arr);
```

Question 5: What is wrong with the following `useEffect` hook, and how would you fix it?

```
useEffect(() => { const timer = setInterval(() => { console.log('Tick'); }, 1000); }, []);
```

Question 6: Write a function that splits a large file into chunks and simulates uploading each chunk sequentially in Node.js.