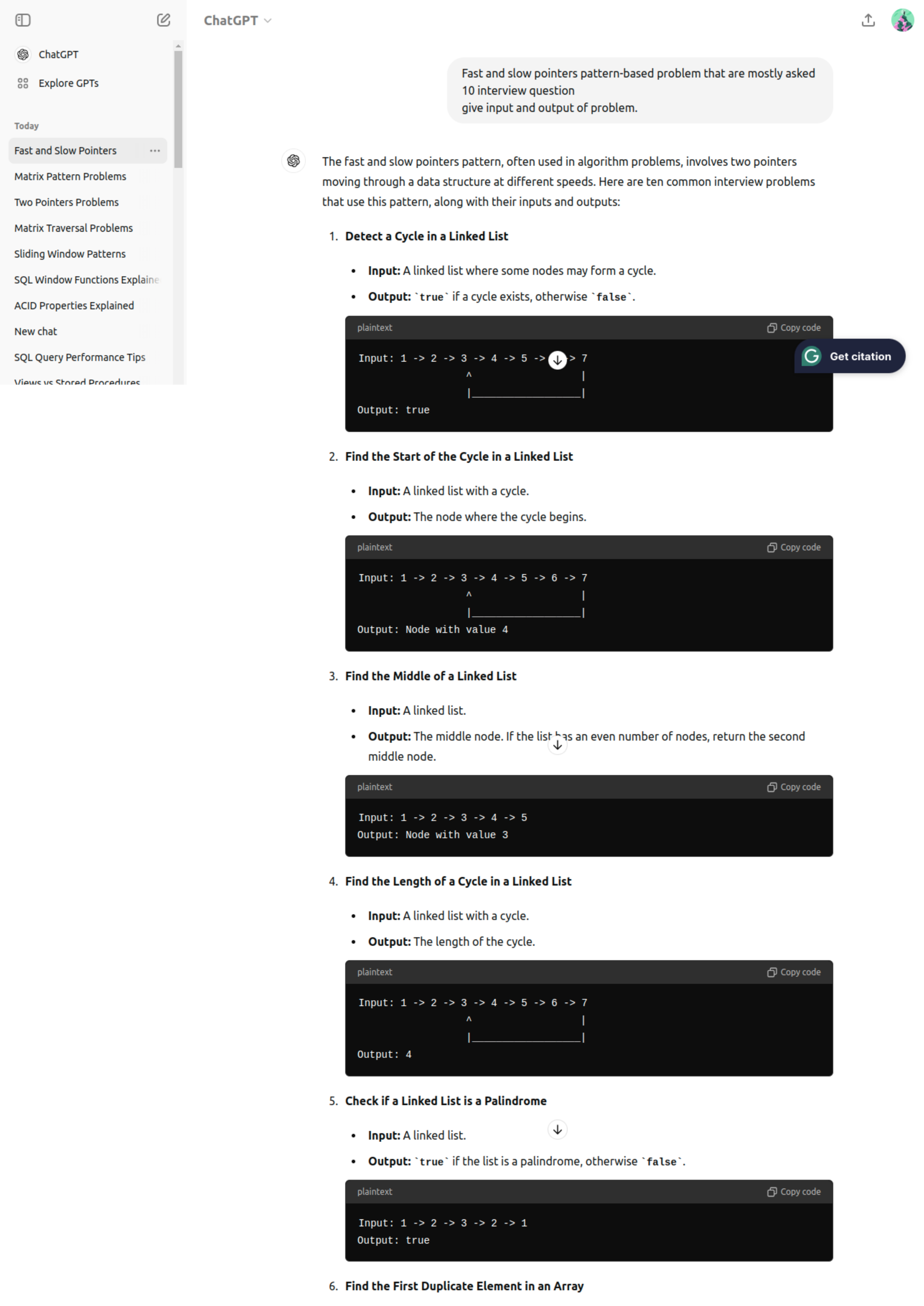

Fast and slow pointers pattern-based problem that are mostly asked 10 interview question
give input and output of problem.

The fast and slow pointers pattern, often used in algorithm problems, involves two pointers moving through a data structure at different speeds. Here are ten common interview problems that use this pattern, along with their inputs and outputs:

1. **Detect a Cycle in a Linked List**
   - **Input:** A linked list where some nodes may form a cycle.
   - **Output:** `true` if a cycle exists, otherwise `false`.

```plaintext
Input: 1 -> 2 -> 3 -> 4 -> 5 ->    > 7
                  ^                 |
                  |_________________|
Output: true
```

2. **Find the Start of the Cycle in a Linked List**
   - **Input:** A linked list with a cycle.
   - **Output:** The node where the cycle begins.

```plaintext
Input: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7
                  ^                 |
                  |_________________|
Output: Node with value 4
```

3. **Find the Middle of a Linked List**
   - **Input:** A linked list.
   - **Output:** The middle node. If the list has an even number of nodes, return the second middle node.

```plaintext
Input: 1 -> 2 -> 3 -> 4 -> 5
Output: Node with value 3
```

4. **Find the Length of a Cycle in a Linked List**
   - **Input:** A linked list with a cycle.
   - **Output:** The length of the cycle.

```plaintext
Input: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7
                  ^                 |
                  |_________________|
Output: 4
```

5. **Check if a Linked List is a Palindrome**
   - **Input:** A linked list.
   - **Output:** `true` if the list is a palindrome, otherwise `false`.

```plaintext
Input: 1 -> 2 -> 3 -> 2 -> 1
Output: true
```

6. **Find the First Duplicate Element in an Array**

- **Input:** An array of integers.
- **Output:** The first duplicate number. If no duplicates, return `-1`.

```plaintext
Input: [2, 1, 3, 4, 2, 5]
Output: 2
```

7. **Find the Duplicate Number (Array with n+1 Elements)**
   - **Input:** An array containing `n + 1` integers where each integer is between `1` and `n`.
   - **Output:** The duplicate number.

```plaintext
Input: [1, 3, 4, 2, 2]
Output: 2
```

8. **Remove Duplicate Characters from a String**
   - **Input:** A string with possible duplicate characters.
   - **Output:** The string with duplicates removed, maintaining original order.

```plaintext
Input: "abacabad"
Output: "abcde"
```

9. **Find the Cycle Length in a Linked List**
   - **Input:** A linked list with a cycle.
   - **Output:** The length of the cycle.

```plaintext
Input: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7
                  ^                 |
                  |_________________|
Output: 4
```

10. **Finding the Intersection Point of Two Linked Lists**
    - **Input:** Two linked lists that may intersect.
    - **Output:** The node where the lists intersect, or `null` if they do not intersect.

```plaintext
Input: List1: 1 -> 2 -> 3 -> 4 -> 5
           List2: 6 -> 4 -> 5
Output: Node with value 4
```

These problems often use the two-pointer technique to achieve an optimal solution with linear time complexity.

Upgrade plan
Get GPT-4, DALL·E, and more

Message ChatGPT

ChatGPT can make mistakes. Check important info.