1. What is a singly linked list, and how does it differ from an array?
   A singly linked list is a type of data structure made up of nodes. Each node has two parts: one stores the data, and the other points to the next node in the list. The list starts with a "head" node and ends with a node that points to null (meaning there's no next node).

The main difference from an array is how the data is stored. Arrays keep elements in a fixed-size block of memory, and you can quickly access any element using its index. Linked lists store elements in separate memory locations and you have to go through the list one node at a time to find something.

2. When would you prefer a linked list over an array, and vice versa?

You'd prefer a linked list when:

- You don't know how many items you'll need to store.
- You need to add or remove items often, especially in the middle.

You'd prefer an array when:

- You know the number of items in advance.
- You need fast access to items using their position (like getting the 5th item quickly).

3. How are linked lists used in real-world applications (e.g., browser history, undo functionality)?

Linked lists are used in many real-life programs. For example:

- Browser history: Each page you visit is a node. You can go back and forward through your history using the links between nodes.
- Undo in apps: Each action is saved as a step. You can undo by going back through the list of actions.
- Music playlists: Songs can be linked one after another, so you play the next song by following the link.

4. Cite your reference/s

- GeeksforGeeks. (n.d.). *Singly Linked List*. https://www.geeksforgeeks.org/data-structures/linked-list/singly-linked-list/

- TutorialsPoint. (n.d.). *Data Structure - Linked List*. https://www.tutorialspoint.com/data_structures_algorithms/linked_list_algorithms.htm

- Cormen, T. H., et al. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.