1. What is a singly linked list, and how does it differ from an array?

   My answer to this is the singly has a first node, its called head. Contains data and a pointer to the next node in the sequence. It's a unidirectional structure. It’s different from array because the singly linked list stores elements in separate memory while connected by pointer, allowing dynamic size and efficient insertions or deletions, but requiring sequential access. But in array it stores element in a contiguous memory, it has a fixed size and slower deletions due to shifting elements.

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

   When I'm adding something, I like to utilize a linked list since you can add a new node without having an impact on the others. For quick or straightforward tasks, I would rather use an array. Due to its ease of management, elements can be accessed by index or number of items.

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

   Because linked lists make updates simple, they are often used in practical applications. to data, like adding or deleting components. For instance, in a song playlist, adding, removing, or rearranging pieces is simple.

4. Cite your reference/s

   Weiss, M. A. (2013). Data Structures and Algorithm Analysis in C++ (4th ed.). Pearson.
   Malik, D. S. (2010). Data Structures Using C++ (2nd ed.). Course Technology.