Exercise Task: Building a Double-Linked List

Create a double-linked list and implement functions for adding, traversing, printing, deleting, editing, and inserting nodes.

Task 1: Define the Double-Linked List Structure

- Define a `Node` structure with the following fields:
- `data` (choose an appropriate data type, like `int` or `char[]`)
- `next` (pointer to the next `Node`)
- `prev` (pointer to the previous `Node`)

Task 2: Implement Basic Operations

1. **Create a Node**: Function to allocate and initialize a new node.
2. **Print List**: Function to traverse and print the data in each node.
3. **Add Node at End**: Function to add a new node at the end of the list.
4. **Add Node at Beginning**: Function to add a new node at the beginning of the list.

Task 3: Advanced Operations

1. **Insert Node After a Given Node**: Function to insert a new node after a specified node.
2. **Delete a Node**: Function to delete a specific node from the list.
3. **Edit a Node's Data**: Function to modify the data of a specific node.
4. **Traverse the List in Both Directions**: Function to traverse the list from beginning to end, and then in reverse order.

Task 4: Test the List

- Write a `main` function to test all the above functionalities.
- Ensure students handle memory allocation and deallocation properly.

Additional Challenge

- Implement a function to sort the list based on the data in the nodes.