Assignment: Linked List Implementation in C/C++

Objective: The objective of this assignment is to implement a sorted linked list in C++ that stores records containing a person's name and age. The program should allow users to perform operations such as adding, deleting, and printing records based on name or age.

Requirements:

1. **Linked List Implementation:**
    - Implement a linked list data structure to store records.
    - Each record should contain two fields: name (a string) and age (an integer).
    - Implement functions for creating a new linked list, adding a record, deleting a record, and printing the list.
2. **Sorted Insertion:**
    - Ensure that the linked list remains sorted alphabetically by name at all times.
    - When adding a new record, insert it into the correct position in the list based on the alphabetical order of names.
3. **Deletion:**
    - Implement functionality to delete any or all records in the linked list.
    - Provide options for deleting records by name or by age.
4. **Printing:**
    - Implement a function to print the linked list in order of insertion (sorted by name).
    - Implement a function to print the linked list in order of age, from youngest to oldest.
5. **User Interface:**
    - Develop a simple command-line interface for users to interact with the program.
    - Provide options for adding new records, deleting records, and printing the list sorted by name or age.

Submission Guidelines:

- Submit a single C++ source file containing the implementation of the linked list and all necessary functions.
- Include comments in your code to explain the purpose of each function and any important logic.
- Ensure that your code follows best practices for readability, error handling, and memory management.

Evaluation Criteria:

- Correctness of the linked list implementation, including sorting and deletion functionality.
- Efficiency of algorithms used for insertion, deletion, and sorting operations.
- Clarity and organization of code.
- User interface design and ease of use.
- Adherence to submission guidelines and assignment requirements.

Additional Notes:

- You may use standard libraries for string manipulation and memory management.
- Test your program thoroughly to ensure it functions correctly in various scenarios, including edge cases.

Rubric

1. **Linked List Implementation (30 points):**
    - 10 points: Linked list data structure is correctly implemented.
    - 10 points: Records contain fields for name (string) and age (integer).
    - 10 points: Functions for creating, adding, deleting, and printing records are correctly implemented.
2. **Sorted Insertion (20 points):**
    - 10 points: New records are inserted into the correct position in the list based on alphabetical order of names.
    - 10 points: Linked list remains sorted after insertions.
3. **Deletion (20 points):**
    - 10 points: Functionality to delete records by name is correctly implemented.
    - 10 points: Functionality to delete records by age is correctly implemented.
4. **Printing (20 points):**
    - 10 points: Function to print the linked list in order of insertion (sorted by name) is correctly implemented.
    - 10 points: Function to print the linked list in order of age (from youngest to oldest) is correctly implemented.
5. **Code Compilation (10 points):**
    - 10 points: Code compiles without errors or warnings.

As always, put the names of those who participated in the programming assignment in the header of the source file and submit your code as a C++ source file on D2L.