Milestone 1 - Doubly Linked List, Due September 10, 2025 4:00 PM Pacific Time

The following files will be provided to you, for completion of your milestone:

- dll_node.h // header file defining dll node structure
- dll_node.cpp // dll node file constructor
- doubly_linked_list.h // header file containing doubly linked list class
- json.hpp // header file for processing json files
- milestone1.json // json file containing test cases and its transactions
- milestone1_config.json // json configuration (properties) file
- outputFile – testcase 1.txt // generated output file format (partial results)
- milestone1.cpp /* cpp file containing main, which does the following:
    - Reads configuration file (json format) to:
        - retrieve inputFile (test case file (json format)
        - retrieve outputFile (text file containing generated output)
        - retrieve errorLogFile (text file containing error messages)
    - process inputFile test cases
    - write output to outputFile */

Write a basic Doubly Linked List implementation, which uses the files listed above, and includes the following in a separate cpp file:

- doubly_linked_list.cpp – implementation file that contains the following methods:

    1. isEmpty - Check if the list is empty
    2. insertAtHead - Adds a new node at the beginning of the list
    3. insertAtTail - Adds a new node at the end of the list
    4. remove - Searches for a node with a specific value and deletes it from the list
    5. removeHeaderNode – removes header node
    6. removeTailNode – removes tail node
    7. moveNodeToHead – moves a specific node to the front
    8. moveNodeToTail – moves a specific node to the end
    9. clear - Clear the list (delete all nodes)
    10. printList - print the doubly linked list from head to tail to console and output file
    11. reversePrintList- print the doubly linked list list from tail to head to console and output file

The total number of points for this milestone is **85**, which will be based upon the following:

- Each submitted/modified file must have student's name (-10% of total milestone points if missing)
- Each submitted file must include a file header with a description of changes made to a program, and its change date (1)
- Program compiles with all of the provided files (1)
- The following methods are documented:

  1. isEmpty - Check if the list is empty (1)
  2. insertAtHead - Adds a new node at the beginning of the list (1)
  3. insertAtTail - Adds a new node at the end of the list (1)
  4. remove - Searches for a node with a specific value and deletes it from the list (1)
  5. removeHeaderNode – removes header node (1)
  6. removeTailNode – removes tail node (1)
  7. moveNodeToHead – moves a specific node to the front (1)
  8. moveNodeToTail – moves a specific node to the end (1)
  9. clear - Clear the list (delete all nodes) (1)
  10. printList - print the doubly linked list (1)
  11. reversePrintList- reverse print the doubly linked list (1)

- The following methods run without errors:

  1. isEmpty - Check if the list is empty (2)
  2. insertAtHead - Adds a new node at the beginning of the list (2)
  3. insertAtTail - Adds a new node at the end of the list (2)
  4. remove - Searches for a node with a specific value and deletes it from the list (2)
  5. removeHeaderNode – removes header node (2)
  6. removeTailNode – removes tail node (2)
  7. moveNodeToHead – moves a specific node to the front (2)
  8. moveNodeToTail – moves a specific node to the end (2)
  9. clear - Clear the list (delete all nodes) (2)
  10. printList - print the doubly linked list (2)
  11. reversePrintList- reverse print the doubly linked list (2)

- The following test cases are processed, and produce expected output (10 per test case; 50 total)
- Extra Credit – use industry standard test program and/or extract test cases, in separate json test file

Please accept this GitHub Assignment:

https://classroom.github.com/a/yvJuJwYB