Lab 5

Linked Lists

Build a **Double Linked Queue** structure that expands as it acquires its nodes (and contracts as it releases its nodes) from a single **Linked Stack** pool of 1000 nodes.

Your Linked Queue should be initially of size zero.

The Queue should employ a Stack initially populated with "clean empty" nodes ready to be leased, and later returned.

Data Structure and methods for Queue:

- A Constructor to create the Linked Queue and Linked Stack of NODELEMENTs.
- Obtain a node from the Stack and Insert an element at the back of the queue.
- REMOVE an element at the front of the queue and release node to the Stack.
- Query any node to show its NODE ID and its information CONTENTS.
- · Test for EMPTY and FULL conditions,
- Show the HEAD and TAIL slot number and contents.
- Show the COUNT of the current elements in the queue (the current size).
- Show the link to the next and previous nodes in the structure.
- Traverse the Queue structure from TAIL to HEAD via the links.
- Traverse the Queue structure from HEAD to TAIL via the links.
- A FIND option that leverages traverse with an information query.
- Show priority INSERT and priority REMOVE capabilities.

Testing via the Main user interface:

Write a program that tests each of the methods used to manipulate the data structure

When displaying the full structure use a loop that shows each slot and its content.

Your program should employ a menu that repeatedly prompts the user to perform the above functions.

Demonstrate that it works for the structure's "boundary conditions".

Output Example of a well-formatted Traverse:

Tail Contents>		53	678	567	456	345	123	99	88	77	< Head Contents
Node Slot	>	4	8	7	6	5	0	1	2	3	< Node Slot
Back Link	>	-1	4	8	7	6	5	0	1	2	< Back Link
Forw Link	>	8	7	6	5	0	1	2	3	-1	< Forw Link