

Tanishq Gadkari
011859770

Part 2 Pseudocode:

stackQueue()

- assign max size

stackQueue(unsigned int)

- assign the given max size

Void enqueue(elemType)

- check to see if its full then throw exception

- if not full then push onto the primary stack

elemType dequeue()

- check if it's empty if empty then throw exception

- loop the primary stack until one element, keep populating the secondary stack

- store the one element in a temp since that's our front and pop that element

- store the elements in the secondary stack back into the primary stack

- return our temp variable that holds the front of the queue

elemType dequeue()

- check if it's empty if empty then throw exception

- loop the primary stack until one element, keep populating the secondary stack

- IMPORTANT take the one element and put it in the temp but don't pop it

- store the elements in the secondary stack back into the primary stack (loop)

- return our temp which is our front

Unsigned int size()

- return size from the primary stack

Unsigned int maxSize()

- return max_size attribute from our class

Bool isEmpty()

- return primary stack's empty method

Bool isFull()

- return the comparison between primary stack's size and max_size attribute

| Method | Time Complexity |
|------------------------|-----------------|
| Void enqueue(elemType) | O(1) |
| elemType dequeue() | O(n) |
| elemType front() | O(n) |
| Unsigned int size() | O(1) |
| Unsigned int maxSize() | O(1) |
| Bool isEmpty() | O(1) |
| Bool isFull() | O(1) |

References:

Exception Handling -

https://www.tutorialspoint.com/cplusplus/cpp_exceptions_handling.htm

STL stack-

<https://en.cppreference.com/w/cpp/container/stack>

Peers:

Kcirde and Garrett