Given the documentation, I wrote a number of tests to try to determine what functions were working properly.

**OUTPUT OF THE TEST SUITE:**

Testing the creator function:

A queue that is created should be empty. PASS!

A queue that is created should be able to be added to. PASS!

Testing the clear entries function:

A queue that is populated then deletes all entries should be empty. FAILED! Was expecting the queue to be empty, but it's not!

Memory leaks should be checked for outside this program.

Testing the isEmpty() function:

An empty (newly created) queue should return false. PASS!

A queue with an item should return false. PASS!

A queue with multiple items should return false. PASS!

A queue that is empty, pushed and popped should return true. PASS!

Testing the Enqueue() function:

Adding an entry to an empty queue should make it not empty. PASS!

Adding an entry to an empty queue should result in that entry being in front. PASS!

Adding two entries should still result in the first entry being in front. FAILED! The entry at the front was not 1 as expected, it was 2!

Adding two entries and popping should result in the second entry being in front. FAILED! The entry at the front was not 2 as expected, it was 1!

Testing the Dequeue() function:

Trying to dequeue an empty stack should throw std::runtime\_error. PASS, it threw an error! If it didn't say PASS just now, then it FAILED.

Removing from a queue of size one should return an empty queue. PASS!

Removing from a queue of size two should make the last entry first. FAILED! The entry at the front was not 2 as expected, it was 1!

Testing the peekFront() function:

Trying to peek an empty stack should throw std::runtime\_error. PASS, it threw an error! If it didn't say PASS just now, then it FAILED.

Adding one entry should return that entry when peeked. PASS!

Adding two entries should return the first entry when peeked. FAILED! The entry at the front was not 1 as expected, it was 2!

Adding two entries and popping should return the second entry when peeked. FAILED! The entry at the front was not 2 as expected, it was 1!

Adding four entries and popping should return the second entry when peeked. FAILED! The entry at the front was not 2 as expected, it was 3!

In addition to the tests formally in the test suite, there were three tests that could not be run in the test suite: the test for memory leaks in the queue clearing function, which was done in Valgrind, did not show any memory leaks. The other two omitted tests were also in the queue clearing function, and were commented out because they threw errors that could not be caught and crashed the program. One tested to see the value of the item at the front of the cleared queue, which I wanted to test because a previous test said it was not empty. Another tested to see if something could be pushed to the cleared queue, which I added to check if it behaved like an empty queue. I also commented out some code to test a theory where this function was deleting the queue, but it resulted in compiling errors, therefore debunking my theory.

**ANALYSIS OF FUNCTIONS:**

THE CREATOR FUNCTION: Everything ran smoothly. An empty queue was created, and queue-specific functions like enqueue() can be performed on it. No failed behaviors.

THE CLEAR ENTRIES FUNCTION: Some tests failed. The failed behaviors are:

* The test that ran through the suite function created a queue, deleted the entries, then checked to see if it was empty. It was not empty.
* Queue-specific functions such as peek and enqueue threw errors.

THE ISEMPTY() FUNCTION: Everything ran smoothly. A newly created queue was empty, a queue with one and another with multiple entries returned false, and a queue that is pushed and popped returned false.

THE ENQUEUE() FUNCTION: Some tests failed. It was able to add entries, and the first entry was in front. However, some failed behaviors are:

* When it pushed two entries, the second entry that was pushed ended up in front rather than the first entry as expected.
* Adding two entries and popping resulted in the first entry being in front, rather than the second entry as expected.

THE DEQUEUE() FUNCTION: Some tests failed. It correctly threw an error when trying to dequeue an empty queue and removing an entry from a queue of length one made the queue empty. However, some failed behaviors are:

* Removing from a queue of size two resulted in the first entry being in front, rather than the second entry as expected.

THE PEEKFRONT() FUNCTION: Some tests failed. It correctly threw an error when peeking an empty function and adding one entry returned a correct value, but some failed behaviors are:

* Adding two entries returned in the second entry being in “front”.
* Adding two entries and popping returned the first entry being in “front”.
* Adding four entries and popping did not return the second entry, it returned the third entry.

**POSSIBLE BUGS:**

**The clear entries function is not working** well at all – it can’t even run queue functions afterwards. I thought it might destroy the queue itself rather than deleting all entries, but this does not make sense, as it should not be able to use isEmpty() on an item that does not exist.

Enqueue(), Dequeue(), and PeekFront() failed some of their tests. They behaved just fine when dealing with one entry in a queue, but things behaved incorrectly when adding more entries. When adding entries to the queue, it was always done in the order of “1”, “2”, “3”, and “4”. What struck me about the error results is that **the queue is behaving like a stack**. Here are the reasons why:

* peekFront() will return the value of the most recent entry added, like a stack. To get it to work like a queue, it needs to return the oldest entry first.
* Dequeue() appears to remove the most recently added entry, like a stack. This behavior is further proven by one of the tests for the peekFront() function! To get it to work like a queue, it needs to remove the oldest entry.
* Enqueue() seems to function properly, despite its failed tests: it entries to the back of the queue, as expected. The errors in its output could be due to the errors in peekFront() and Dequeue() that were used in the tests.