

```
1: /*Copyright 2017 Sam Pickell*/
2: #define BOOST_TEST_DYN_LINK
3: #define BOOST_TEST_MODULE Main
4: #include <boost/test/unit_test.hpp>
5: #include <string>
6: #include <exception>
7: #include "RingBuffer.hpp"
8:
9: BOOST_AUTO_TEST_CASE(RBconstructor) {
10:     // normal constructor
11:     BOOST_REQUIRE_NO_THROW(RingBuffer(100));
12:
13:     // this should fail
14:     BOOST_REQUIRE_THROW(RingBuffer(0), std::exception);
15:     BOOST_REQUIRE_THROW(RingBuffer(0), std::invalid_argument);
16: }
17:
18: BOOST_AUTO_TEST_CASE(RBenque_dequeue) {
19:     RingBuffer rb(3);
20:
21:     rb.enqueue(2);
22:     rb.enqueue(1);
23:     rb.enqueue(0);
24:
25:     BOOST_REQUIRE(rb.dequeue() == 2);
26:     BOOST_REQUIRE(rb.dequeue() == 1);
27:     BOOST_REQUIRE(rb.dequeue() == 0);
28:
29:     BOOST_REQUIRE_THROW(rb.dequeue(), std::runtime_error);
30: }
31:
32: BOOST_AUTO_TEST_CASE(RBpeek) {
33:     RingBuffer rb(3);
34:
35:     rb.enqueue(42);
36:     rb.enqueue(100);
37:     rb.enqueue(9001);
38:
39:     // Can't enqueue full buffer
40:     BOOST_REQUIRE_THROW(rb.enqueue(1), std::runtime_error);
41:
42:     // Check peek
43:     BOOST_REQUIRE(rb.peek() == 42);
44:
45:     BOOST_REQUIRE(rb.dequeue() == 42);
46:     BOOST_REQUIRE(rb.dequeue() == 100);
47:     BOOST_REQUIRE(rb.dequeue() == 9001);
48:
49:     BOOST_REQUIRE_THROW(rb.dequeue(), std::runtime_error);
50:     // Can't peek into an empty buffer
51:     BOOST_REQUIRE_THROW(rb.peek(), std::runtime_error);
52: }
```