

```
1: // Copyright 2015 <Angel Z'heondre Calcano>
2: // PS5a
3: #include <stdint.h>
4: #include <stdexcept>
5: #include <iostream>
6: #include <string>
7: #include <vector>
8: #include "RingBuffer.hpp"
9:
10: int RingBuffer::size() { return _currentcapacity; }
11:
12: bool RingBuffer::isEmpty() { return _buffer.empty(); }
13:
14: bool RingBuffer::isFull() {
15:     if (_buffer.size() == (unsigned)_size)
16:         return true;
17:     if (_buffer.size() > (unsigned)_size)
18:         return false;
19:     else
20:         return false;
21: }
22:
23: void RingBuffer::enqueue(int16_t x) {
24:     if (_currentcapacity == _size)
25:         throw
26:             std::runtime_error("Can't enqueue on a full ring");
27:     _buffer[_last] = x;
28:     _last++;
29:     _currentcapacity++;
30:     if (_last == _size)
31:         _last = 0;
32: }
33:
34: int16_t RingBuffer::dequeue() {
35:     if (_currentcapacity <= 0)
36:         throw
37:             std::runtime_error(" Can't dequeue from empty ring");
38:     double _hold;
39:     _hold = _buffer[_first];
40:     _first++;
41:     _currentcapacity--;
42:     if (_first == _last) _first = _last = 0;
43:     if (_first == _size) _first = 0;
44:     return _hold;
45: }
46:
47: int16_t RingBuffer::peek() {
48:     if (_currentcapacity == 0)
49:         throw
50:             std::runtime_error("Can't peek on an empty ring");
51:     if (_buffer.empty())
52:         throw
53:             std::runtime_error("Can't peek on an empty vector array");
54:     return _buffer[_first];
55: }
```