```
 1: // Copyright 2015 <Angel Zheondre Calcano>
 2: // PS5b
 3: #ifndef _GuitarString_
 4: #define _GuitarString_
 5: #include <math.h>
 6: #include <limits.h>
 7: #include <SFML/System.hpp>
 8: #include <stdint.h>
 9: #include <cstdlib>
10: #include <iostream>
11: #include <string>
12: #include <exception>
13: #include <stdexcept>
14: #include <vector>
15: #include "RingBuffer.hpp"
16:
17: class GuitarString{
18:   RingBuffer *_j; int _size, _ticCount;
19:
20:  public:
21:   explicit GuitarString(double freq) {
22:     if (freq < 1)
23:       throw std::runtime_error("Constructor frequency must be > than 0");
24:     _size = ceil((48400/freq)); //48400
25:     _j = new RingBuffer(_size);
26:     _ticCount = 0;
27:      for (int i = 0 ; i < _size; i++)
28:        _j->enqueue(0);
29:   }
30:   explicit GuitarString(std::vector< sf::Int16 > j) {
31:     _size = j.size();
32:     if (_size < 1)
33:       throw std::runtime_error("Empty Vector, Size must be > than 0 ");
34:     _j = new RingBuffer(_size);
35:     _ticCount = 0;
36:     for (int i = 0; i < _size; i++) {
37:       _j->enqueue((int16_t)j[i]);
38:     }
39:   }
40:   ~GuitarString() {
41:     delete _j; // delete *_j made it fail the test.
42:   }
43:   void pluck();
44:   int time();
45:   void tic();
46:   sf::Int16 sample();
47: };
48: #endif
```