```
 1: /*Copyright Sam Pickell 2017*/
 2: #include "GuitarString.hpp"
 3: #include <cmath>
 4: #include <cstdlib>
 5: #include <vector>
 6:
 7: GuitarString::GuitarString() {
 8: }
 9:
10: GuitarString::GuitarString(double frequency) {
11:   int N = ceil(44100/frequency);
12:   data = new RingBuffer(N);
13:   for (int i = 0; i < N; i++) {
14:       data->enqueue(0);
15:     }
16: }
17:
18: GuitarString::GuitarString(std::vector<sf::Int16> init) {
19:   data = new RingBuffer(init.size());
20:   for (unsigned int i = 0; i < init.size(); i++) {
21:       data->enqueue(init.at(i));
22:     }
23: }
24:
25: GuitarString::~GuitarString() {
26:   delete data;
27: }
28:
29: void GuitarString::pluck() {
30:   unsigned int r = 123;
31:   for (int i = 0; i < data->size(); i++) {
32:       int16_t random_var = (int16_t)(rand_r(&r) % 0xffff);
33:       data->dequeue();
34:       data->enqueue(random_var);
35:     }
36: }
37:
38: void GuitarString::tic() {
39:   int16_t KS_update = data->peek();
40:   data->dequeue();
41:   KS_update = ((KS_update + data->peek()) / 2) * .996;
42:   data->enqueue(KS_update);
43:
44:   tic_tracker++;
45: }
46:
47: sf::Int16 GuitarString::sample() {
48:   return data->peek();
49: }
50:
51: int GuitarString::time() {
52:   return tic_tracker;
53: }
```