```
 1: // Copyright [2015] Justin Nguyen
 2:
 3: #include <iostream>
 4: #include <cmath>
 5: #include <string>
 6: #include "LFSR.hpp"
 7:
 8: LFSR::LFSR(std::string seed, int t) {
 9:   _data = seed;
10:   length =  _data.size();
11:   tap = t;
12: }
13:
14: int LFSR::step() {
15:   int bit;
16:   int length = _data.size();
17:   std::string s_bit;
18:   bit = _data.front() ^ _data[length - tap - 1];
19:   s_bit = std::to_string(bit);
20:   _data.erase(0, 1);
21:   _data = _data + s_bit;
22:   return bit;
23: }
24:
25: int LFSR::generate(int k) {
26:   int count = 0;
27:   for (int i = k - 1; i >= 0; i--) {
28:     if (step() == 1)
29:       count +=  pow(2, i);
30:   }
31:   return count;
32: }
33:
34: std::ostream& operator << (std::ostream& out, LFSR& lfsr) {
35:   out << lfsr._data;
36:   return out;
37: }
```