```
 1: // Copyright 2015 <Angel Z'heondre Calcano>
 2: // PS2a
 3: #include <iostream>
 4: #include <string>
 5: #include <vector>
 6: #include "LFSR.hpp"
 7:
 8: using namespace std;
 9:
10: int LFSR::stTodig(std::string &a) {
11:   char b; int d1, d2;
12:   b = a[0];
13:   if (b == '1') d1 = 1;
14:   else
15:     d1 = 0;
16:   b = a[a.length() - t - 1];
17:   if (b == '1') d2 = 1;
18:   else
19:     d2 = 0;
20:   return (d1 ^ d2);
21: }
22: int LFSR::step() {
23:   int bit; char c; char digits[3] = { '0', '1', '\0' };
24:   bit = stTodig(unit);
25:   c = digits[bit];
26:   unit.erase(unit.begin());
27:   unit.push_back(c);
28:   return bit;
29: }
30: int LFSR::generate(int k) {
31:   int i, num, total,count;
32:   vector< int > reg;
33:   reg.resize(k);
34:   total = count =0;
35:   num = 1;
36:   for(i = 0; i < k; i++) {
37:     reg[i] = step();
38:   }
39:   for ( i = k - 1; -1 < i; i--) {
40:     if( i != k - 1 )
41:       num = num * 2;
42:     if( reg[i] == 1){
43:       total += num;
44:     };
45:   }
46:   return total;
47: }
48: std::string LFSR::prtln(const LFSR &temp, int bit) {
49:   std::cout << temp << " " << bit;
50:   return "";
51: }
52:
53:
54:
```