```
1: // Copyright 2015 < Angel Z'heondre Calcano>
 2: // PS2a
 3: #include <iostream>
 4: #include <string>
 5: #include "LFSR.hpp"
7: using namespace std ;
8:
9: int LFSR::stTodig( string &a ) {
10:
    char b; int d1, d2;
11:
    b = a[0] ;
12:
13:
    if(b == '1') d1 = 1;
    else d1 = 0;
14:
15:
16: b = a[a.length() - t - 1];
17: if (b == '1') d2 = 1;
18: else d2 = 0;
19:
20:
    return d1 ^ d2 ;
21: }
22: //unit[unit.length() - t - 1]
23: int LFSR::step(){
    char c ;
24:
25:
    //stTodig(untit)
26:
    bit = stTodig( unit ) ;
27: c = (char)bit;
28: unit.erase(unit.begin());
29: unit.append(1,c);
30:
     return bit ;
31: }
32:
33: int LFSR::generate( int k ) {
34: // calls step k times. on the kth time take
35: //the string and take the k amount of bits
36: // and return it's value
37:
38:
    int i, num, total ;
39:
40:
    total = 0 ;
41:
42:
     for( i = 0; i < k; i++)
43:
      step();
44:
     for( i = 0 ; i < k ; i++ ) {
45:
       if(i == 0) num = 1;
46:
       else num = num * num ;
47:
       if( unit[i] == '1' ) total = total + num ;
48:
49:
     return total ;
50: }
51: /*
52: std::ostream& operator<< ( std::ostream &out, LFSR& rhs ) {
54:
     out << rhs.unit ;
    return out ;
55:
56: }
57: */
58: string LFSR::prtln(const LFSR &temp, int bit ) {
    cout << temp << " " << bit ;
59:
60: }
61:
```

LFSR.cpp Thu May 07 06:46:08 2015 2

62:

63: