```
 1: // Copyright 2015 <Angel Z'heondre Calcano>
 2: // PS2b
 3: #include <iostream>
 4: #include <string>
 5: #include "LFSR.hpp"
 6:
 7: using namespace std ;
 8:
 9: int LFSR::stTodig( string &a ) {
10:    char b; int d1, d2 ;
11:
12:    b = a[0] ;
13:    if( b == '1' ) d1 = 1 ;
14:    else d1 = 0 ;
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(){
24:    char c ;
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 ) {
53:
54:    out << rhs.unit ;
55:    return out ;
56: }
57: */
58: string LFSR::prtln(const LFSR &temp, int bit ) {
59:    cout << temp << " " << bit ;
60: }
61:
```

```
62:
63:
```