```
// Copyright 2015 <Angel Z’heondre Calcano>
// PS2a
#include <iostream>
#include <string>
#include "LFSR.hpp"

using namespace std ;

int LFSR::stTodig( string &a ) {
  char b; int d1, d2 ;

  b = a[0] ;
  if( b == ’1’ ) d1 = 1 ;
  else d1 = 0 ;

  b = a[a.length() - t - 1] ;
  if( b == ’1’ ) d2 = 1;
  else d2 = 0 ;

  return d1 ^ d2 ;
}
//unit[unit.length() - t - 1]
int LFSR::step(){
  char c ;
  //stTodig(untit)
  bit = stTodig( unit ) ;
  c = (char)bit ;
  unit.erase(unit.begin()) ;
  unit.append(1,c) ;
  return bit ;
}

int LFSR::generate( int k ) {
// calls step k times. on the kth time take
//the string and take the k amount of bits
// and return it’s value

  int i, num, total ;

  total = 0 ;

  for( i = 0 ; i < k ; i++ )
    step() ;
  for( i = 0 ; i < k ; i++ ) {
    if( i == 0 ) num = 1 ;
    else num = num * num ;
    if( unit[i]== ’1’ ) total = total + num ;
  }
  return total ;
}
/*
std::ostream& operator<< ( std::ostream &out, LFSR& rhs ) {

  out << rhs.unit ;
  return out ;
}
*/
string LFSR::prtln(const LFSR &temp, int bit ) {
  cout << temp << " " << bit ;
}

```