```
/*
Computing IV - Assignment - PS1a + b
Instructor: Prof. Yelena Rykalova
Due Date: 02/07/22
Author: Matthew Lorette Anaya
Description: This program is an implementation of a Fibonacci Linear Feed
back               Shift Register
             Takes in a seed and generates bits with seed() and numbers w
ith g              enerate(int)
*/
#include <string>
#include <sstream>
#include <math.h>
#include "FibLFSR.h"

FibLFSR::FibLFSR(std::string seed) {
  int size = seed.length();
  // No try-catchblock for BOOST test
  if(size != 16)
    throw std::invalid_argument("Incorect seed bit length, must be 16.");
  reg = seed;
}

int FibLFSR::getBit(char a) {
  if (a == '1') return 1;
  else if (a == '0') return 0;
  else return 1;
}

int FibLFSR::xOr(int a, int b) {
  return a != b;
}

std::ostream& operator<<(std::ostream& os, FibLFSR &lfsr) {
  os << lfsr.reg;

  return os;
}

int FibLFSR::step() {

  //new register after shifting
  std::string new_reg = reg.substr(1);

  //Taps(10, 12, and 13)
  //{Equal = 0}{Not Equal = 1}
  int tap = xOr(reg[0], reg[2]);
  tap = xOr(tap, getBit(reg[3]));
  tap = xOr(tap, getBit(reg[5]));

  FibLFSR::reg = new_reg;
  FibLFSR::reg += std::to_string(tap);

  return tap;
}

int FibLFSR::generate(int k) {
  int result = 0;
  for(int i = 0; i < k; i++){
    int z = step();
    result = (result * 2) + z;
  }

  return result;
}
```