```
    1: /**
    2:  * @file test.cpp
    3:  * @author Shivam Patel
    4:  * @brief
    5:  * Course: Comp 4
    6:  * Due Date: 2022-01-31
    7:  * @date 2022-01-31
    8:  *
    9:  * @copyright Copyright (c) 2022
   10:  *
   11:  */
   12: // Dr. Rykalova
   13: // test.cpp for PS1a
   14: // updated 1/31/2020
   15:
   16: #include <iostream>
   17: #include <string>
   18:
   19: #include "FibLFSR.h"
   20:
   21: #define BOOST_TEST_DYN_LINK
   22: #define BOOST_TEST_MODULE Main
   23: #include <boost/test/unit_test.hpp>
   24:
   25: BOOST_AUTO_TEST_CASE(sixteenBitsThreeTaps) {
   26:
   27:   FibLFSR l("1011011000110110");
   28:   BOOST_REQUIRE(l.step() == 0);
   29:   BOOST_REQUIRE(l.step() == 0);
   30:   BOOST_REQUIRE(l.step() == 0);
   31:   BOOST_REQUIRE(l.step() == 1);
   32:   BOOST_REQUIRE(l.step() == 1);
   33:   BOOST_REQUIRE(l.step() == 0);
   34:   BOOST_REQUIRE(l.step() == 0);
   35:   BOOST_REQUIRE(l.step() == 1);
   36:
   37:   FibLFSR l2("1011011000110110");
   38:   BOOST_REQUIRE(l2.generate(9) == 51);
   39: }
   40:
   41: //This test case tests our getter function and makes sure that the seed i
s getting updated after stepping.
   42: //compared the value returned by getSeed (a string) with the value that s
hould be returned to make sure the seed gets updated after each step
   43: BOOST_AUTO_TEST_CASE(getterTest) {
   44:   FibLFSR l2("1011011000110110");
   45:   l2.step();
   46:   BOOST_REQUIRE(l2.getSeed() == "0110110001101100");
   47:   l2.step();
   48:   BOOST_REQUIRE(l2.getSeed() == "1101100011011000");
   49:   l2.step();
   50:   BOOST_REQUIRE(l2.getSeed() == "1011000110110000");
   51: }
   52:
   53: //This tests my unstringify fucntion which converts a string passed in as
 either a '0' or  '1', to an integer 0 or 1.
   54: //Unstrigify was used in my step function.
   55: BOOST_AUTO_TEST_CASE(unstringifyTest) {
   56:  char bit0 = '0';
   57:  BOOST_REQUIRE(unstringify(bit0) == 0);
   58:  char bit1 = '1';
   59:  BOOST_REQUIRE(unstringify(bit1) == 1);
   60: }
```