```
    1: /**
    2: * FibLFSR.cpp - To run the FubLFSR class, in which they constructor, step
(),
    3: * and generator(), essentially using Linear feedback shift register
    4: *
    5: * Date 1/24/22 - 1/31/22
    6: *
    7: * Created by: Anson Cheang
    8: *
    9: */
   10:
   11: #include "FibLFSR.h"
   12: #include <vector>
   13: #include <string>
   14: #include <cmath>
   15:
   16: using namespace std;
   17:
   18: FibLFSR::FibLFSR(string seed)
   19: {
   20:     string Pbit;
   21:     for(int i = 0; i < static_cast <int> (seed.length()); i++)
   22:     {
   23:         Pbit = seed[i];
   24:         list.push_back(stoi(Pbit, 0, 10));
   25:
   26:     }
   27: }
   28:
   29: int FibLFSR::step()
   30: {
   31:     int size = static_cast <int> (list.size());
   32:     int RBit = list[0] ^ list[2] ^ list[3] ^ list[5];
   33:     for(int i = 0; i < size - 1; i++)
   34:     {
   35:         list[i] = list[i+1];
   36:     }
   37:
   38:     list[size - 1] = RBit;
   39:
   40:     return RBit;
   41: }
   42:
   43: int FibLFSR::generate(int k)
   44: {
   45:     if(k > 32 || k < 0)
   46:     {
   47:         throw out_of_range("The value inputted isn't within the range of
0 - 32");
   48:     }
   49:     int RBit;
   50:     int total = 0;
   51:     for(int i = 0; i < k; i++)
   52:     {
   53:         RBit = step();
   54:         total = total + pow(2, k-i-1) * RBit;
   55:     }
   56:
   57:     return total;
   58: }
   59:
   60: ostream& operator<<(ostream& out, FibLFSR CurrentBits)
   61: {
   62:     vector<int>::iterator it;
   63:     for(it = CurrentBits.list.begin(); it != CurrentBits.list.end(); it++
```

```
)
    64:     {
    65:         out << *it;
    66:     }
    67:     return out;
    68: }
```