

```
1: /**
2:  * @file FibLFSR.cpp
3:  * @author Shivam Patel
4:  * @brief ps1 - definitions for FibLFSR member functions
5:  * Due Date: 2022-01-31
6:  * Course: Comp 4
7:  * @date 2022-01-31
8:  *
9:  */
10: #include "FibLFSR.h"
11:
12: int unstringify(char bit){
13:
14:     if(bit == '0'){
15:         bit = 0;
16:     }else if(bit == '1'){
17:         bit = 1;
18:     }
19:     return bit;
20: }
21:
22: ostream& operator<<(ostream& out, const FibLFSR& l){
23:     out << l.getSeed() << " ";
24:     return out;
25: }
26:
27:
28: FibLFSR::FibLFSR(string seed){
29:     iseed = seed;
30: }
31:
32: string FibLFSR::getSeed() const{
33:     return iseed;
34: }
35:
36: int FibLFSR::step(){
37:
38:     int counter;
39:     int bit;
40:     //extract the tap bits that we need
41:     char bit0 = iseed.at(0);
42:     char bit2 = iseed.at(2);
43:     char bit3 = iseed.at(3);
44:     char bit5 = iseed.at(5);
45:
46:     //convert the string values to integer values
47:     bit = unstringify(bit0);
48:     counter = unstringify(bit2);
49:     //xor the first 2 tap postitions
50:     counter = counter ^ bit;
51:
52:     //convert from string to integer
53:     bit = unstringify(bit3);
54:     //xor the tap bits
55:     counter = counter ^ bit;
56:
57:     //convert from string to integer
58:     bit = unstringify(bit5);
59:     //xor the tap bits
60:     counter = counter ^ bit;
61:
62:     //append the return value to the string and remove the first bit
63:     iseed.append(to_string(counter));
64:     iseed.erase(0, 1);
65:
```

```
66:     return counter;
67: }
68:
69: int FibLFSR::generate(int k){
70:
71:     const int accum = 2;
72:     int bin2num = 0;
73:     string val;
74:     //append the values returned by step k number of times to a string th
ats k bits long.
75:     for(int i = 0; i < k; i++){
76:         val.append(to_string(this->step()));
77:     };
78:
79:     //convert the generated binary string to a decimal value and return
80:     int j = 0;
81:     for(int x = k - 1; x > 0; x--){
82:         if(val.at(x) == '1'){
83:             bin2num += pow(accum, j);
84:         }
85:         j++;
86:     }
87:     return bin2num;
88: }
```