

```

#include <iostream>
#include <vector>

int main()
{
    std::bitset<32> b1;
    std::bitset<32> b2;
    int iterations = 0;
    //Get the 2 binary strings that we want to do LFSR with
    std::cout << "Enter a 32-bit Binary Number: ";
    std::cin >> b1;
    std::cout << "Enter another 32-bit Binary Number: ";
    std::cin >> b2;
    //Gets the amount of bits for the returned binary string
    std::cout << "How many iterations? ";
    std::cin >> iterations;

    std::vector<int> ans(iterations);

    //Creates the xor variables for the feedback functions
    int b1_xor = 0;
    int b2_xor = 0;

    //Implementation of the cipher without clocking
    for (int i = ans.size() - 1; i >= 0; i--)
    {
        //We will have xor operand (feedback function) for b1 be b1[15] xor
        b1[28] xor b1[3]
        b1_xor = b1[15] ^ b1[28] ^ b1[3];
        //We will have xor operand (feedback function) for b2 be b2[30] xor
        b2[16] xor b2[5]
        b2_xor = b2[30] ^ b2[16] ^ b2[5];

        //Sets the bit of the returned binary string to the xor of the
        feedback functions
        ans[i] = b1_xor ^ b2_xor;

        //Moves binary string b1 over 1 and sets the new last bit to what the
        feedback function returned
        b1 <<= 1;
        b1[0] = b1_xor;

        //Moves binary string b2 over 1 and sets the new last bit to what the
        feedback function returned
        b2 <<= 1;
        b2[0] = b2_xor;
    }

    // Prints the binary string that was made from the LFSR
    for (int i = 0; i < ans.size(); i++)
    {

```

```
        std::cout << ans[i];  
    }  
  
    std::cout << std::endl;  
    return 0;  
}
```