作业一：33 级 LFSR 的实现

学号：　　姓名：

要求：

1. 实现以下伪代码
2. 实现 33 级 LSFR
3. 输出从第 66 个时钟周期后开始
4. 连续输出 64 比特

```
U8 shiftRegister=0x11; /*u8：无符号8比特整型*/
U8 PolyNomial=0x8c,feedback,temp;
long int j=0;

while(j<maxiteration){
    temp = shiftRegister&PolyNomial;
    temp = (temp&0xF)^((temp>>4)&0xF);
    temp = (temp&0x3)^((temp>>2)&0x3);
    feedback = (temp&0x1)^((temp>>1)&0x1);

    bin output=shiftRegister&0x1; /*bin：无符号1比特整型*/
    shiftRgeister=(shiftRegister>>1)|(feedback<<7);
    j=j++;
}
```

大致思路：

1. LSFR 为二进制比特操作，因此选择调用$C++$中的$bitset$类，$bitset$模板类由若干个位（bit）组成，使程序员不必通过位运算就能很方便地访问、修改其中的任意一位。$bitset$类中有多个比特操作的函数，足够满足本题需求

   $bitset$类大致介绍：
   实例化：$bitset<N> name = InitialValue;$
   右移：$name.operator >>= (RightShiftNum);$
   $bitset$实例中 1 的个数：$name.cout();$

2. 按照伪代码的含义给出代码即可

代码：

```
#include <iostream>
#include <bitset>
#define N 33
#define maxiteration 130       // 66 + 64

using namespace std;

int main()
{
    bitset<N> shiftRegister = 0b000100010001000100010001000100010;     // 初始化寄存器的值
    bitset<N> PolyNomial = 0b000110011001100110011001100110011;       // 初始化多项式的值

    long int i = 0;

    while(i < maxiteration)
    {
        if(i >= 66)
            cout << shiftRegister[0] << endl;   // LFSR 运行 66 个 clock 后,输出寄存器最右端的值
        //cout << shiftRegister.to_string() << endl;

        bitset<N> temp = shiftRegister & PolyNomial;    // 多项式与寄存器的值相与
        shiftRegister.operator>>= (1);      // 寄存器中值右移 1 位
        int feedback = temp.count() % 2;    // 计算 temp 中 1 的个数,个数取余 2，即求出 feedback
        shiftRegister[N - 1] = feedback;    // 更新寄存器最左端的值

        i ++ ;
    }

    cout << "此时寄存器中的值: " << shiftRegister.to_string() << endl;

    return 0;
}
```

如代码中所示，
将寄存器初值设置为：00010001000100010001000100010010
多项式的值设置为：00011001100110011001100110011011

对寄存器进行总计 130 次迭代，从第 67 次开始输出寄存器右移以前的最低位。对于 feedback 的计算，伪代码中的多次相与可以**等价为 temp 中 1 的个数对 2 的余数**，因此只需通过等价运算计算出 feedback 后填入右移后寄存器的最左端即可。

该程序的输出为：
0 0 1 0 1 0 0 0 0 0 0 1 1 1 1 1 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 1 1 1 1
1 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 0 1 1 0

最后的寄存器状态：10111100001000110010101111000010 0