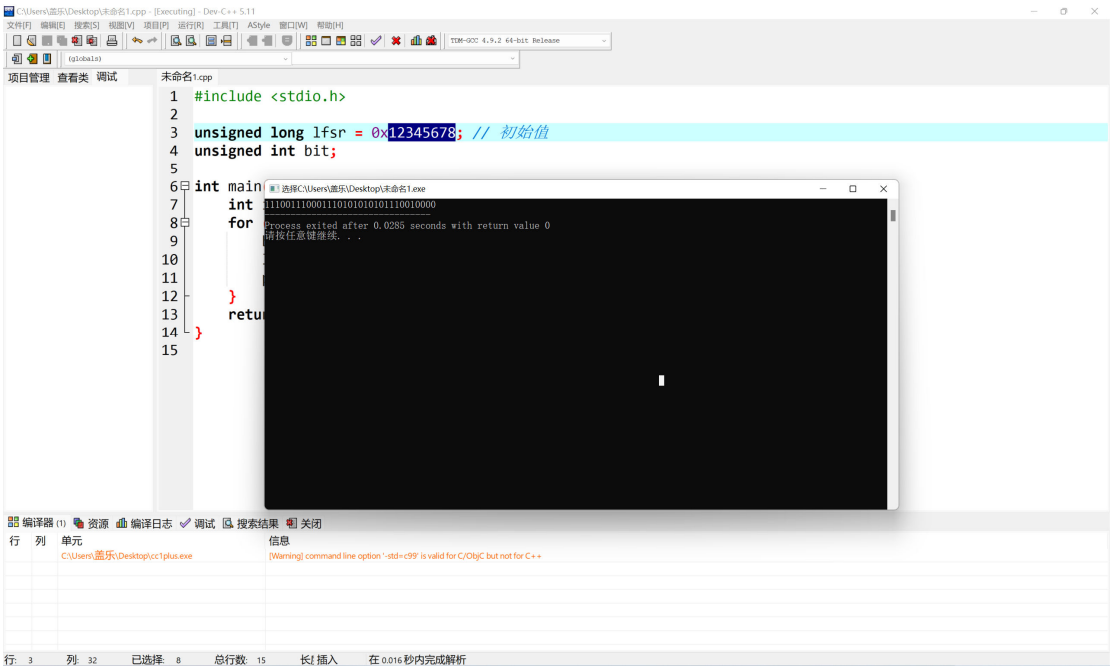
33 位线性反馈移位寄存器 C 语言实现：

其中初始值为 0x12345678；
多项式为 x^0+x^2+x^3+x^31.

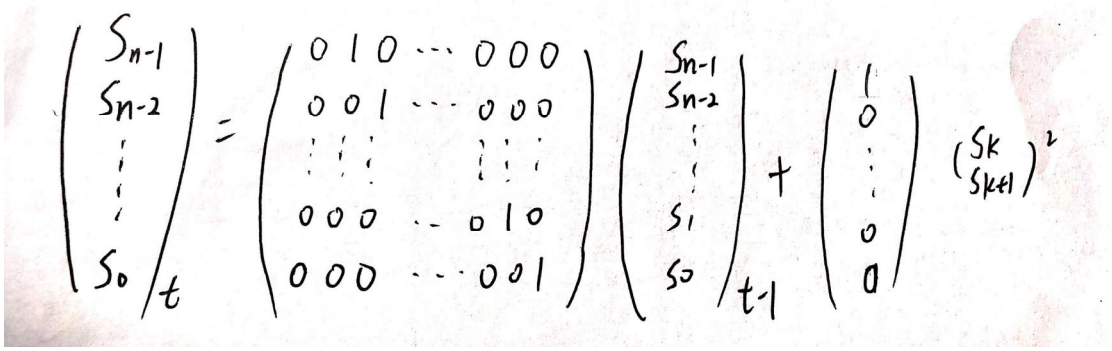
```
#include <stdio.h>

unsigned long lfsr = 0x12345678; // 初始值
unsigned int bit;

int main() {
    int i;
    for (i = 0; i < 33; i++) {
        bit = ((lfsr >> 0) ^ (lfsr >> 2) ^ (lfsr >> 3) ^ (lfsr >> 31))
& 1; // 抽头
        lfsr = (lfsr >> 1) | (bit << 31); // 更新状态
        printf("%d", bit); // 输出
    }
    return 0;
}
```



首先，定义了一个 unsigned long 类型的变量 lfsr 并初始化为 0x12345678。这个变量将用来存储 LFSR 的状态。
然后，定义了一个 unsigned int 类型的变量 bit，这个变量将用来存储 LFSR 的抽头。
在 main 函数中，使用了一个 for 循环，循环次数为 33 次。在每次循环中，首先通过一个表达式来抽取 LFSR 的抽头。这个表达式是：
bit = ((lfsr >> 0) ^ (lfsr >> 2) ^ (lfsr >> 3) ^ (lfsr >> 31)) & 1;

这个表达式中，使用了位运算符 >> 和 ^ 来实现了 LFSR 的抽头。其中，>> 用来移位，^ 用来异或。最后，使用 & 1 来取最低位。这个表达式的结果就是 LFSR 的抽头。
接着，更新 LFSR 的状态。lfsr = (lfsr >> 1) 将 LFSR 的状态右移一位，即舍弃最低位。然后，使用 << 31 来将抽头左移到最高位。最后，使用 | 来将两部分结合在一起，得到新的 LFSR 状态。最后，输出 LFSR 的抽头。

对于含二次交叉项的 LFSR。其矩阵表达式如下：



其中，$s_0, s_1, \ldots, s_{n-1}$ 为 LFSR 的状态向量，n 为 LFSR 的阶数，k 为二次交叉项的位置。
矩阵表达式中的第一个矩阵运算，表示移位操作，使得状态向量中的元素向右移动一位。第二个矩阵运算表示异或操作，即将 s_k 与 s_{k+1} 的平方作为输入，异或到最后一位。