第 1 关：基本测试

根据 S-AES 算法编写和调试程序，提供 GUI 解密支持用户交互。输入可以是 16bit 的数据和 16bit 的密钥，输出是 16bit 的密文。

1.1 截图展示：

说明：在输入框内写入 16bit 数据，选择“二进制”单选按钮后输入任意 16bit 密钥，选中“加密”或“解密”单选框后，点击“立即执行”便得到相应结果。

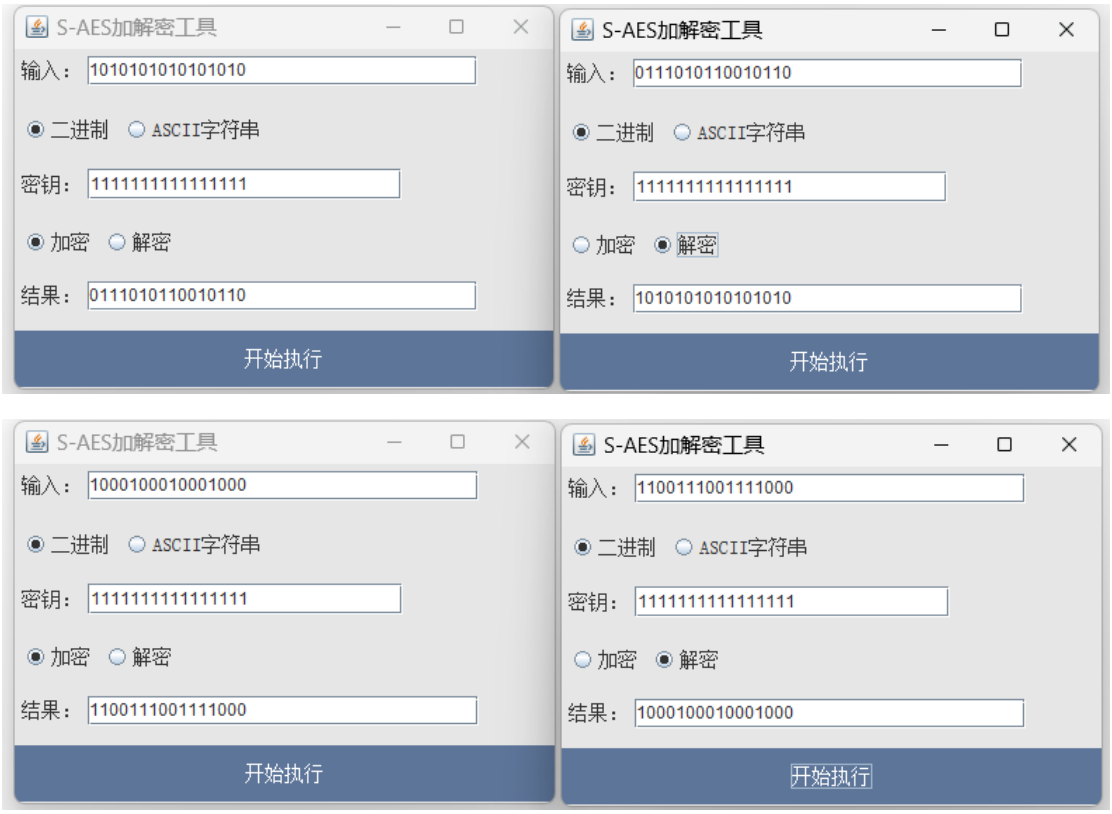


图 1 两对加密交互图示

1.2 数据表格：

表 1：S-AES 明密文对

明文	密钥	密文
1111111111111111	0000000011111111	1010000000100010
0000000000000000	1111111111111111	0000100011000001
0000000011111111	0000000011111111	0111010110011010
1111111100000000	1111111100000000	0100011110100001

第 2 关：交叉测试

考虑到是"算法标准"，所有人在编写程序的时候需要使用相同算法流程和转换单元(替换盒、列混淆矩阵等)，以保证算法和程序在异构的系统或平台上都可以正常运行。设有 A 和 B 两组位同学(选择相同的密钥 K)；则 A、B 组同学编写的程序对明文 P 进行加密得到相同的密文 C；或者 B 组同学接收到 A 组程序加密的密文 C，使用 B 组程序进

行解密可得到与 A 相同的 P。

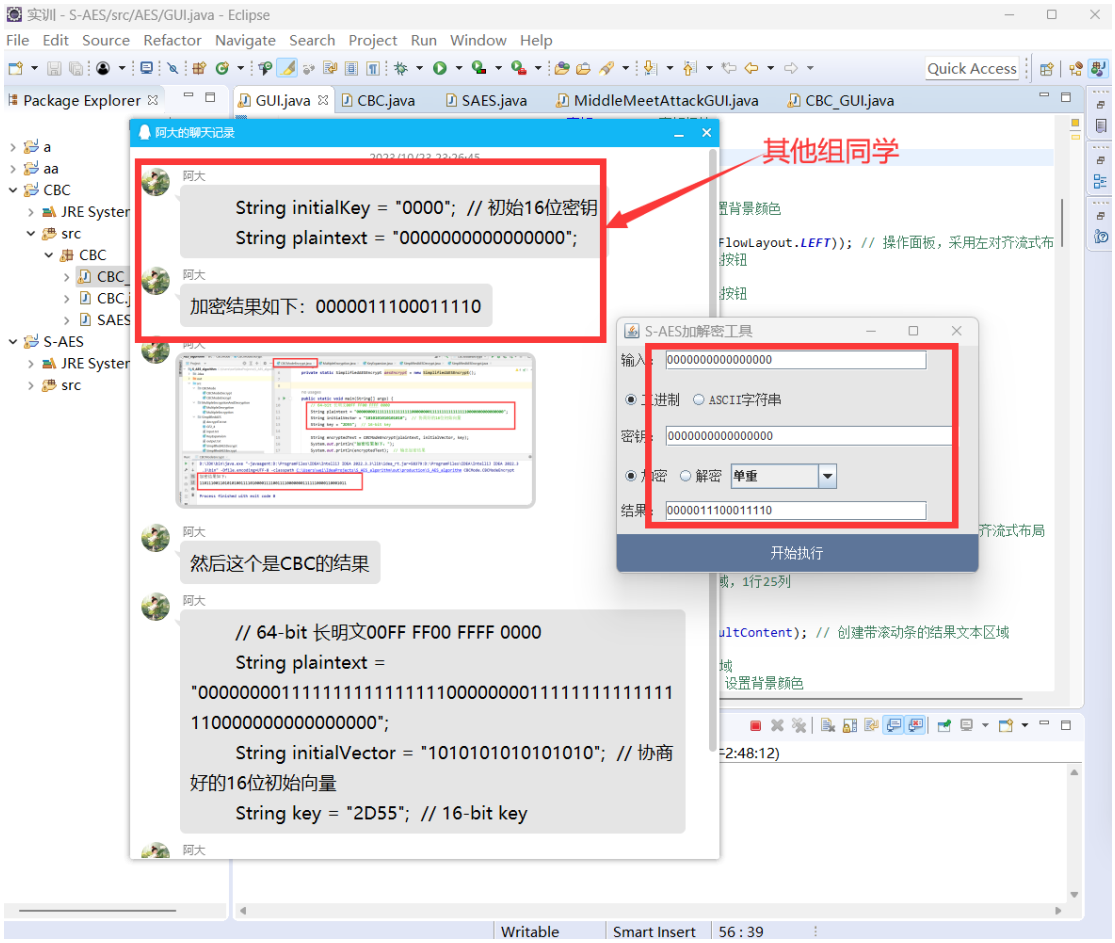


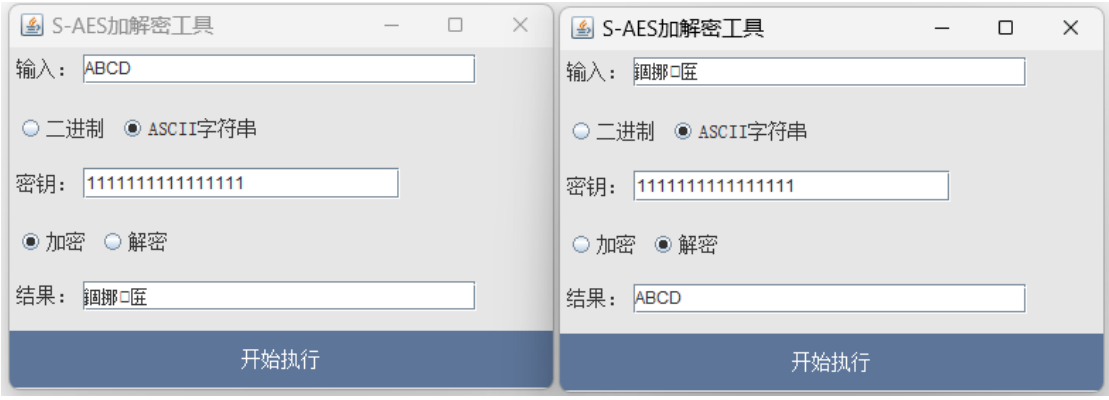
图 2 交叉测试

第 3 关：扩展功能

考虑到向实用性扩展，加密算法的数据输入可以是 ASII 编码字符串(分组为 2 Bytes)，对应地输出也可以是 ACII 字符串(很可能是乱码)。

说明：在输入框内写入 ASII 编码字符串，选择“ASII 字符串”单选按钮后输入任意 10bit 密钥，选中“加密”或“解密”单选框后，点击“立即执行”便得到相应结果。

3.1 截图展示：



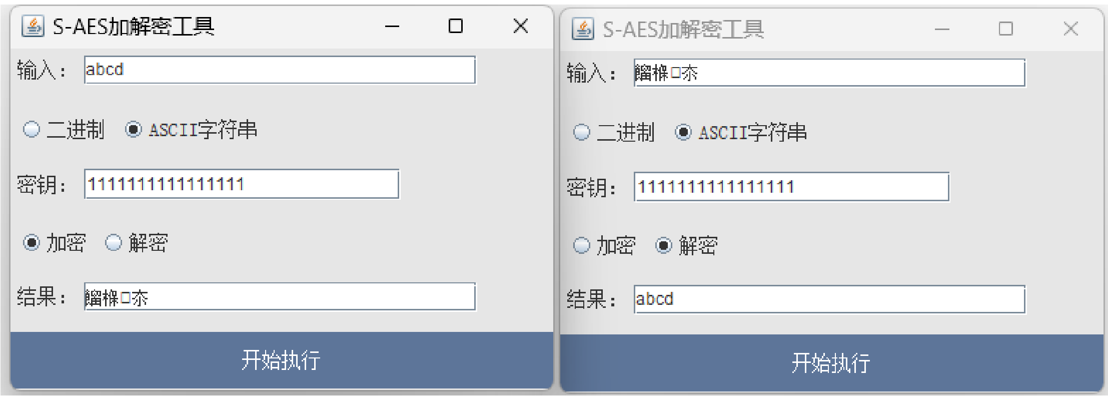


图 3 两对解密交互图示

3.2 数据表格：

表 2： S-AES 明密文对(ASCII)

明文	密钥	密文
重庆大学	1111111111111111	y嗇
我是美女	1111111111111111	噿了縝狥
cqur	1111111111111111	퀜鵭欧氚
cqur	0000000000000000	8ᅥ廜좔祕

第 4 关：多重加密

4.1 双重加密将 S-AES 算法通过双重加密进行扩展，分组长度仍然是 16 bits，但密钥长度为 32 bits。

4.1.1 截图表示

说明：我们将双重加密分为两种模式：①（K1 加密+K2 解密）②（K1 加密+K1 加密）

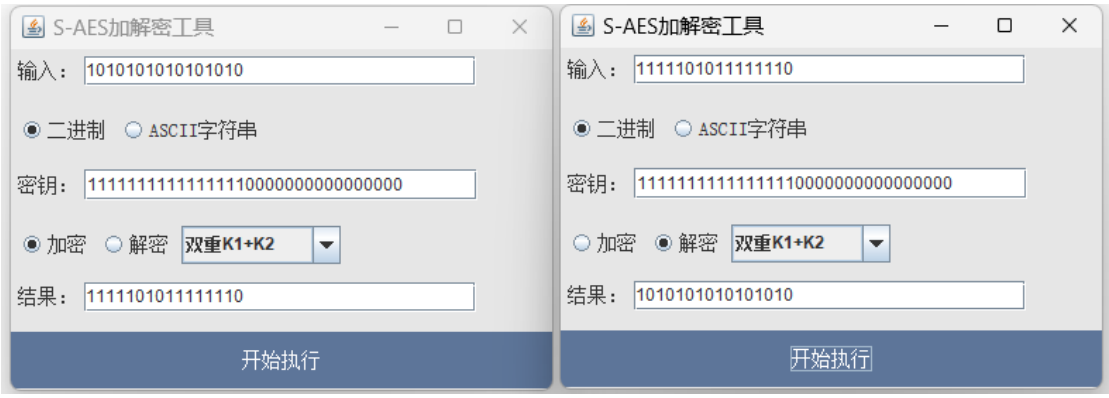


图 4 两对解密交互图示

4.1.2 数据表格

表 3：双重 K1 模式（K1 加密两次）

明文	密钥（16 位）	密文
1111111111111111	1111111111111111	0111101010110101
0000000011111111	1111111111111111	1000011100111001
1111111100000000	1111111111111111	1010001100011010
1111111100000000	0000000000000000	0001000101100011

表 4：双重 K1+K2 模式（K1 加密+K2 解密）

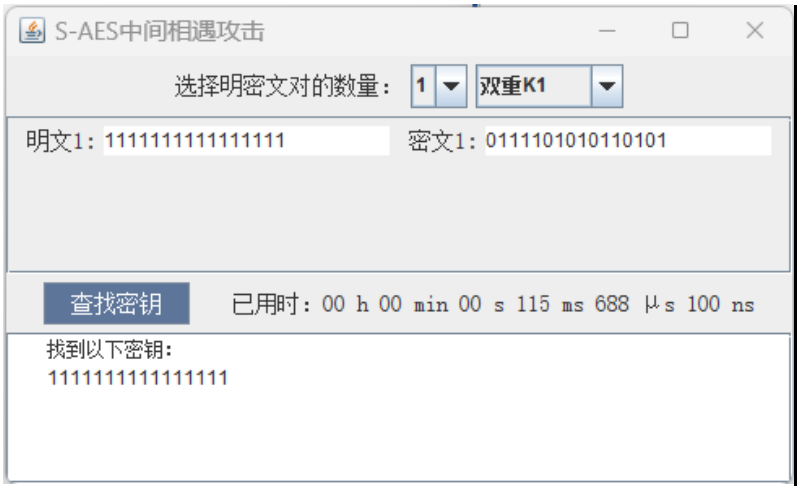
明文	密钥（32 位）	密文
0000000011111111	00000000111111110000000011111111	0000000011111111
0000000011111111	00000000000000001111111111111111	1001010011101110
1111111100000000	11111111111111110000000000000000	1000111101100011
1111111100000000	00000000000000001111111111111111	1011010111001101

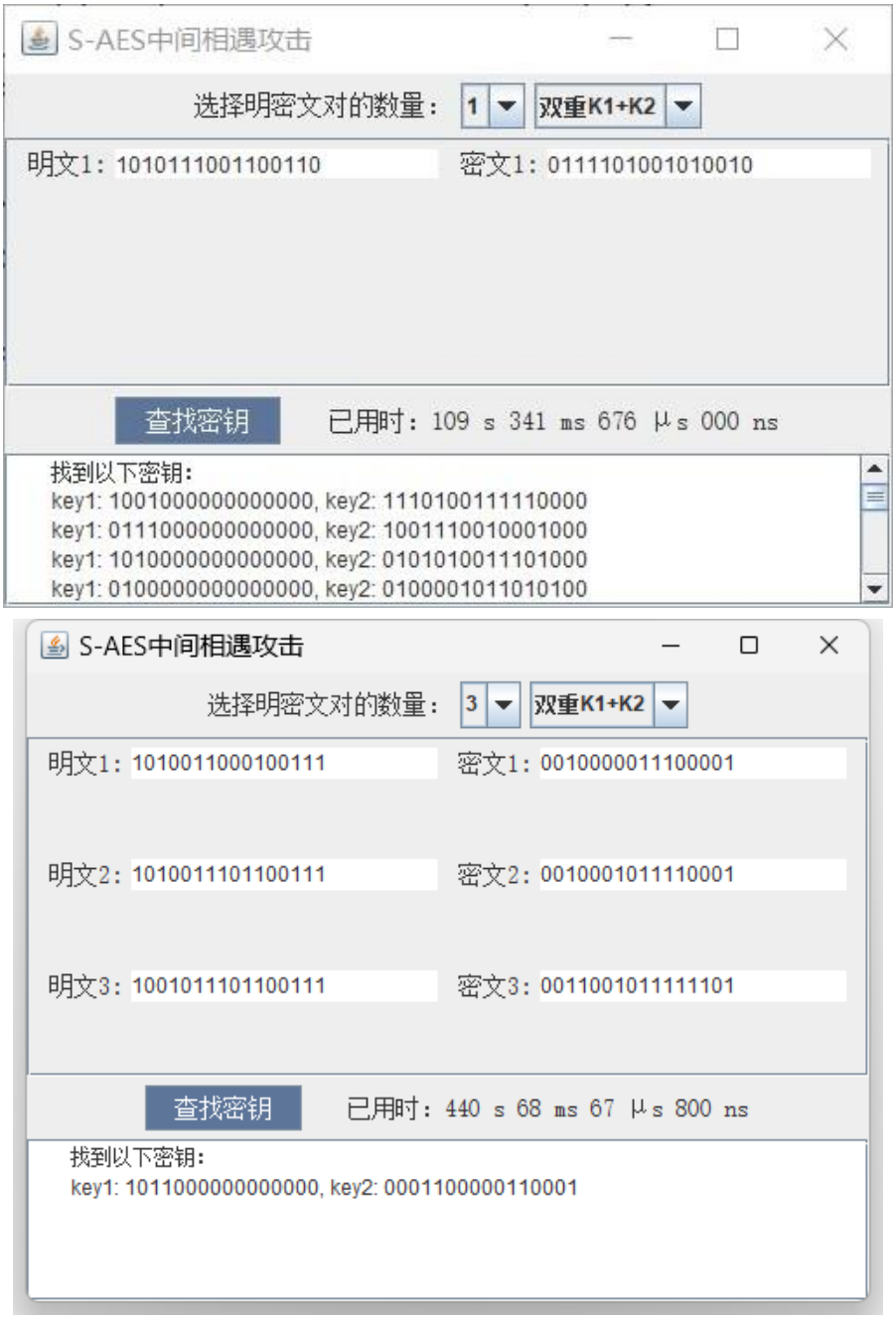
4.2 中间相遇攻击假设你找到了使用相同密钥的明、密文对(一个或多个)，请尝试使用中间相遇攻击的方法找到正确的密钥 Key(K1+K2)。

4.2.1 截图表示

说明：点击下拉框，选择已知明密文对的数量和密钥模式。

（考虑到中间相遇攻击算法破解密钥 K1，K2 时，当 K1=K2 则算法时间复杂度相对较小，毫秒级内可破解，反之则需要分钟级时长。故有“双重 K1”模式和“双重 K1+K2”模式。）





·图 5 中间相遇攻击交互图示

结论：当只有一对明密文时，通过中间相遇攻击算法可以得到多对密钥。当输入多对明密文时，可以逐渐确定唯一的密钥。

4.3 三重加密将 S-AES 算法通过三重加密进行扩展，下面两种模式选择一种完成：
(1)按照 32 bits 密钥 Key(K1+K2)的模式进行三重加密解密，
(2)使用 48bits(K1+K2+K3)的模式进行三重加解密。

4.3.1 截图表示

图 6 两种模式下加解密交互图

4.3.2 数据表格

表 5：32 bits 密钥 Key(K1 加密+K2 解密+K1 加密)的模式

明文	密钥	密文
0000000011111111	00000000000000001111111111111111	0000001101101011
1111111100000000	00000000000000001111111111111111	0111001100110111
1111111111111111	00000000000000001111111111111111	0111010100100010
0000000000000000	00000000000000001111111111111111	0111101111111010

表 6：48 bits 密钥 Key(K1 加密+K2 解密+K3 加密)的模式

明文	密钥	密文
1111111111111111	111111111111111111111111 11111111 1111111111111111	0101001101000011
1111111111111111	000000000000000000000000 000000000000000000000000	0010100100110000
1111111100000000	000000000000000000000000 000000000000000000000000	1101000101000011
1111111100000000	111111111111111111111111 11111111 1111111111111111	0001011000000000

第 5 关：工作模式

基于 S-AES 算法，使用密码分组链(CBC)模式对较长的明文消息进行加密。注意初

始向量(16 bits) 的生成，并需要加解密双方共享。在 CBC 模式下进行加密，并尝试对密文分组进行替换或修改，然后进行解密，请对比篡改密文前后的解密结果。

5. 1 CBC 模式下进行加解密

5.1.1 截图表示

说明：初始 IV 使用随机数生成，下列三组数据 IV 不同

图 7　32 位加解密交互图

图 8　48 位加解密交互图

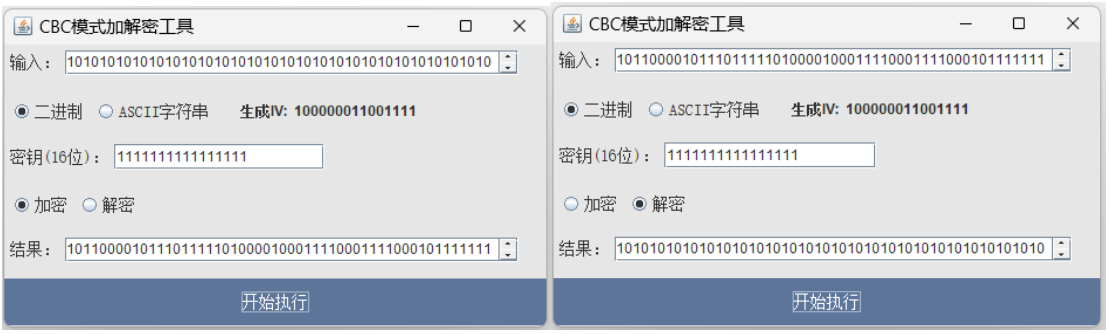


图 9　64 位加解密交互图

5.1.2 数据表格

表 7 以密钥 0xFFFF，IV 0x819F 为例：

明文	密文
1010101010101010 1010101010101010	10110000101110111 110100001000111
1010101010101010 1010101010101010 1010101010101010	1011000010111011 1110100001000111 1000111100010111
1010101010101010 1010101010101010 1010101010101010 1010101010101010	1011000010111011 1110100001000111 1000111100010111 1111011100111111

5. 2 修改密文比对结果

5.2.1 截图表示（以 48 位为例）

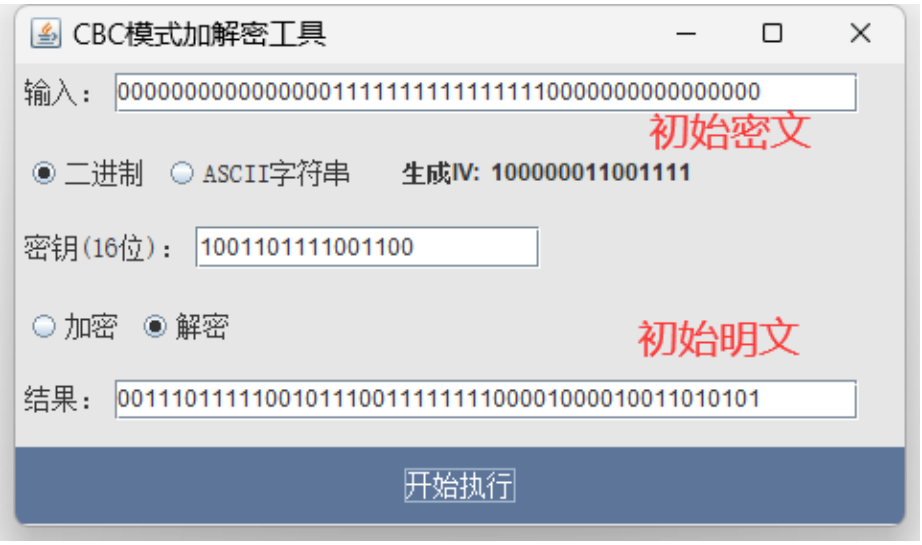


图 10 初始明密文（修改前）

图 11 修改密文进行解密

5.2.1 数据表格

表 8 CBC 模式下的加解密结果

明文	密文
001110111110010111001111	000000000000000011111111
111100001000010011010101	111111110000000000000000
011100000101011000110000	111111110000000011111111
111100001000010011010101	111111110000000000000000
000001000111010011001111	000000001111111111111111
000011111000010011010101	111111110000000000000000
000001000111010001000100	000000001111111100000000
010001000111101111010101	111111110000000000000000