DES算法程序设计和实现

**DES算法原理描述：**

数据加密标准（Data Encryption Standard, DES)是在美国IBM公司的研究人员Horst Feistel 和Walter Tuchman于20世纪70年代中期提出的一个密码算法（LUCIFER算法）的基础之上发展而来的，并于1977年1月15日由美国国家标准局（NBS）正式公布实施，是第一公开的商用密码标准，并得到了ISO的认可。

DES 是一种典型的块加密方法：它以64位为分组长度，64位一组的明文作为算法的输入，通过一系列复杂的操作，输出同样64位长度的密文。

DES 使用加密密钥定义变换过程，因此算法认为只有持有加密所用的密钥的用户才能解密密文。

DES 采用64位密钥，但由于每8位中的最后1位用于奇偶校验，实际有效密钥长度为56位。密钥可以是任意的56位的数，且可随时改变。其中极少量的数被认为是弱密钥，但能容易地避开它们。所有的保密性依赖于密钥。

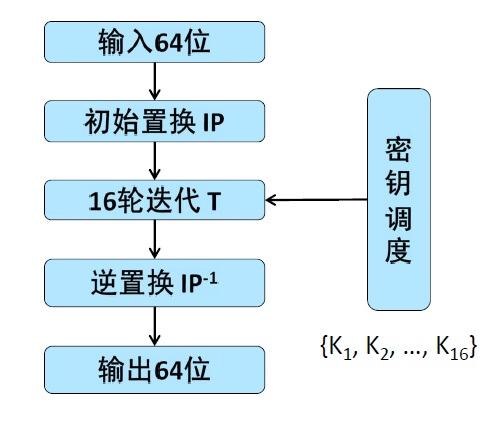
DES 算法的基本过程是换位和置换。

DES算法加密的基本的实现思路如下：

1. 输入明文，进行IP初始置换。
2. 输入密钥，生成16个子密钥。
3. 进行16轮迭代变换。
4. 左右交换后进行逆初始置换（ IP-1 ）得到密文。

解密的基本思路也差不太多，这里就不累述了。

算法的总体结构如下：



具体功能分为四大模块：  
  1. 辅助功能：完成字符和二进制之间的相互转换与异或等功能

2. 密码函数Feistel：完成DES算法轮变换

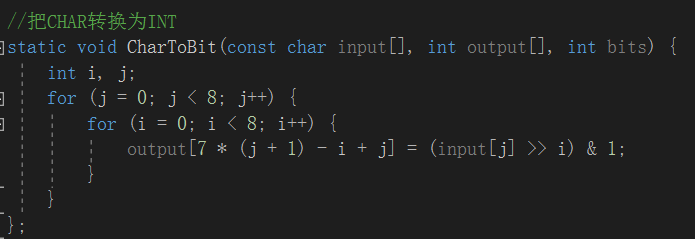
3. 子密钥生成：完成子密钥的生成  
  4. 加密功能：完成DES加密  
  5. 解密功能：完成DES解密

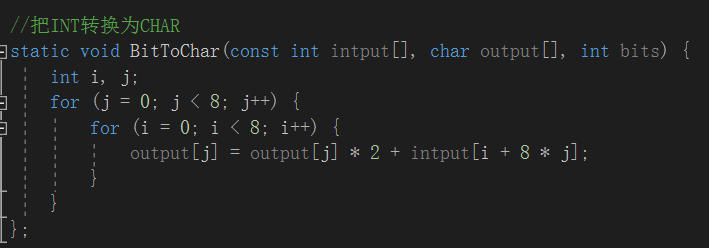
**模块分解：**

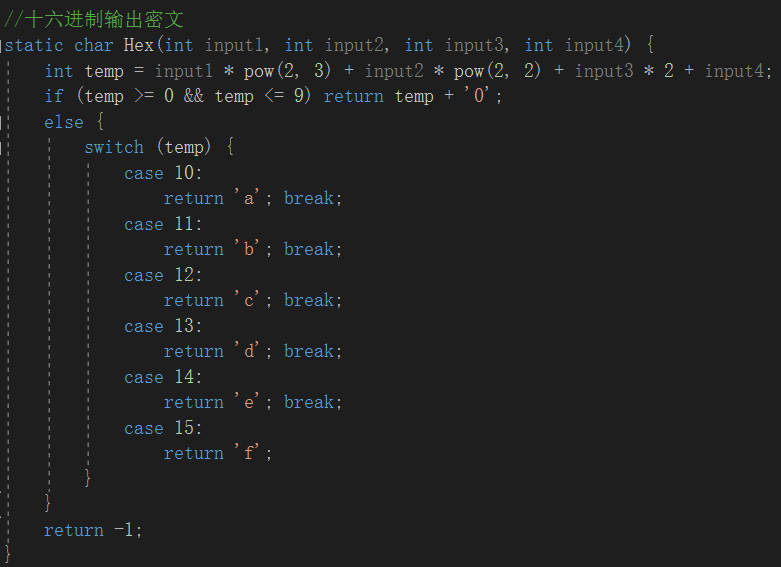
首先是各个DES表的声明，具体的值ppt上已经给出：

|  |  |
| --- | --- |
| 表名称 | 数据结构 |
| IP初始置换表 | 长度为64的一维数组 |
| IP逆置换表 | 长度为64的一维数组 |
| P盒置换表 | 长度为32的一维数组 |
| E盒扩展表 | 长度为48的一维数组 |
| PC1置换表 | 长度为56的一维数组 |
| PC2置换表 | 长度为48的一维数组 |
| S盒（S1~S8） | [8][4][16]的三维数组 |

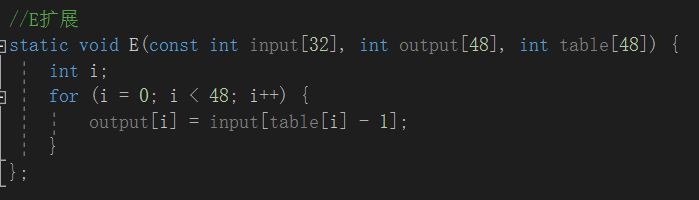
然后是一些辅助函数的声明于实现：



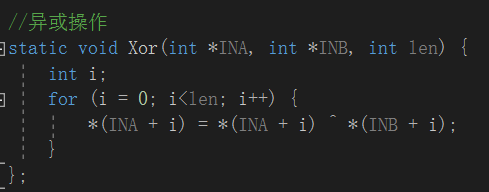




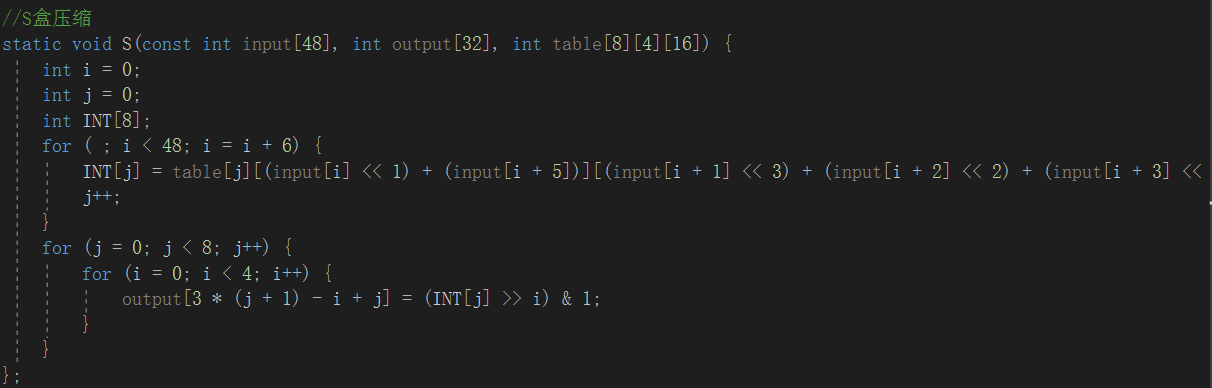
密码函数Feistel：首先，将长度为32位的串Ri-1 作E-扩展，成为48位的串E(Ri-1)；



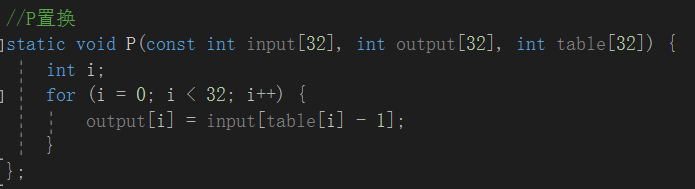
将E(Ri-1) 和长度为48位的子密钥Ki 作48位二进制串按位异或运算，Ki 由密钥K 生成：



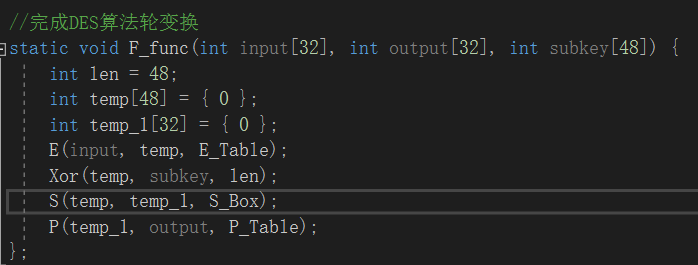
将之前得到的结果平均分成8个分组(每个分组长度6位)，各个分组分别经过8个不同的S-盒进行6-4 转换，得到8个长度分别为4位的分组，得到的分组结果顺序连接得到长度为32位的串；



将得到的结果经过P-置换，作为轮函数f(Ri-1, Ki) 的最终32位输出：



根据ppt上面的给出的S盒，E盒扩展以及P置换规则可以写出相应的函数，最后声明F函数，调用之前写好的方法：



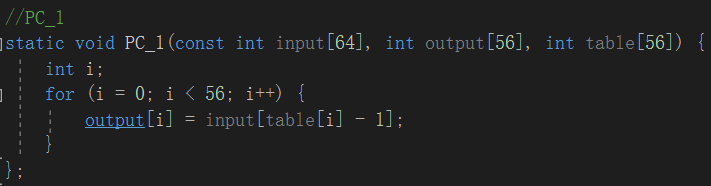
子密钥生成：子密钥生成过程根据给定的64位密钥K 生成Feistel 轮函数的每

轮中使用的子密钥Ki。

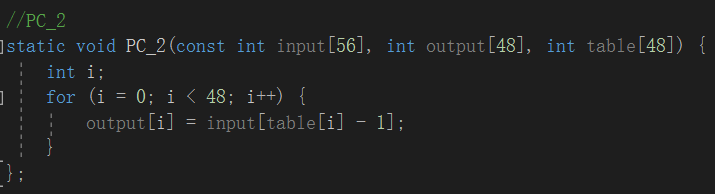
生成子密钥的函数声明如图：



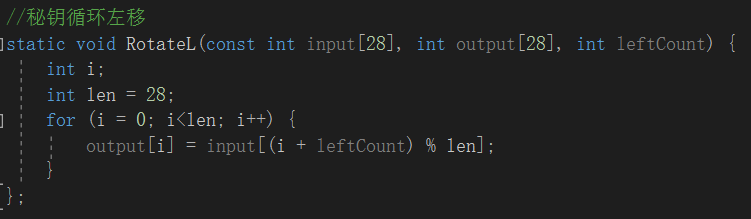
PC-1置换的实现：



PC-2置换的实现：

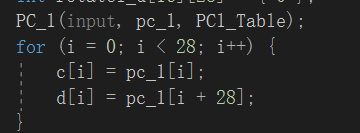


循环左移函数的实现：

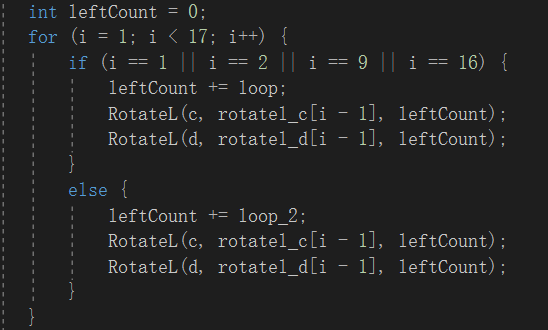


对每轮中使用的子密钥Ki的56个非校验位实行PC-1置换，得到C0D0，其中C0 和D0

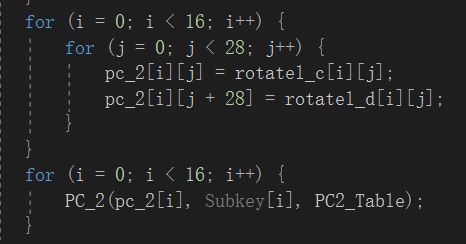
分别由PC-1 置换后的前28位和后28位组成：



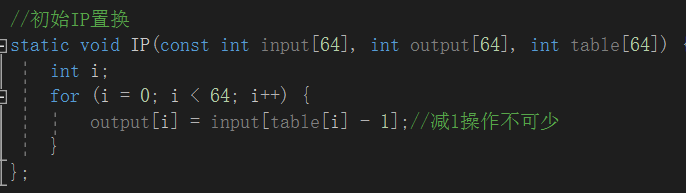
计算Ci = LSi(Ci-1) 和Di = LSi(Di-1)。当i =1, 2, 9, 16 时，LSi (A) 表示将二进制串A 循环左移一个位置；否则循环左移两个位置：

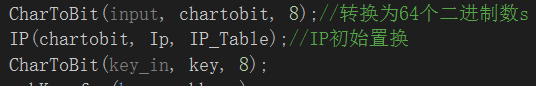


对56位的CiDi 实行PC-2 压缩置换，得到48位的Ki：

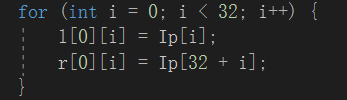


DES加密：首先进行初始IP置换：

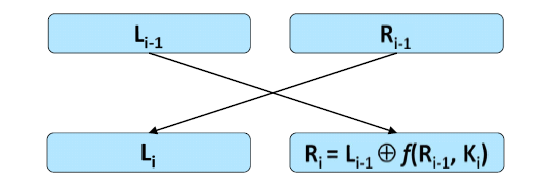


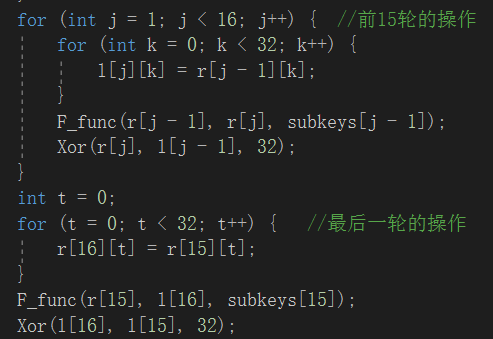


T迭代：先将传入的经IP混淆过的64位明文分为左右两部分，分别为L0（32位）、R0（32位）：

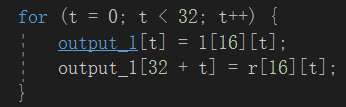


然后经过以下转换关系，进行16轮迭代，得到L16R16：





将L16和R16交换，得到R16L16，作为IP逆置换的输入：

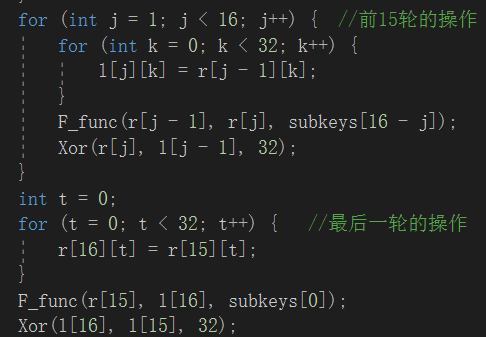


最后进行逆置换IP，得到最终密文：



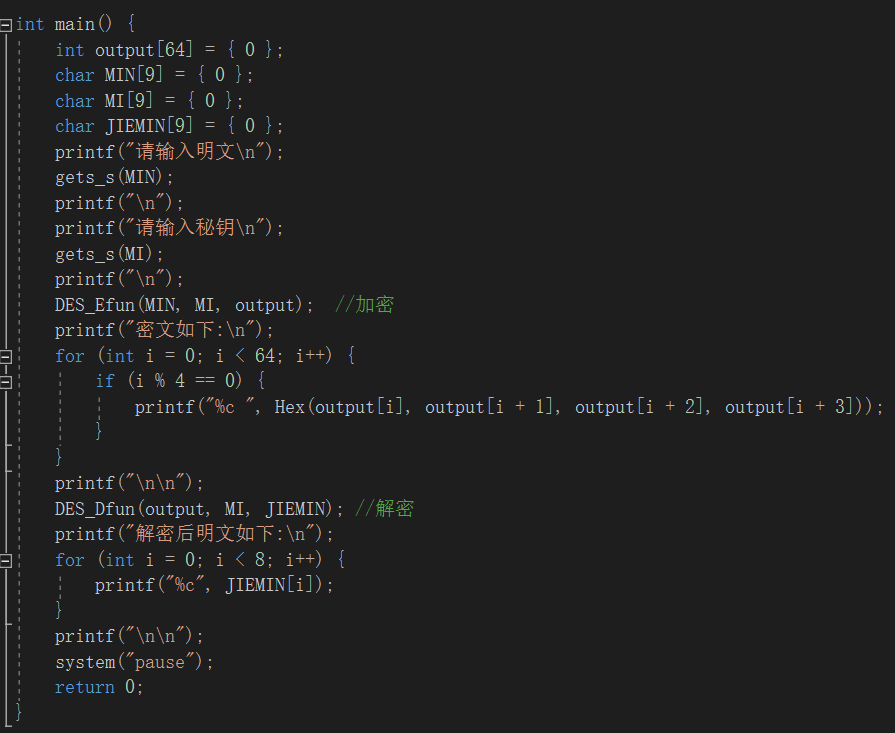
DES解密：与加密不同的地方是，密钥调度时子密钥按(K16K15 … K1)

次序调度，其余地方基本相同：

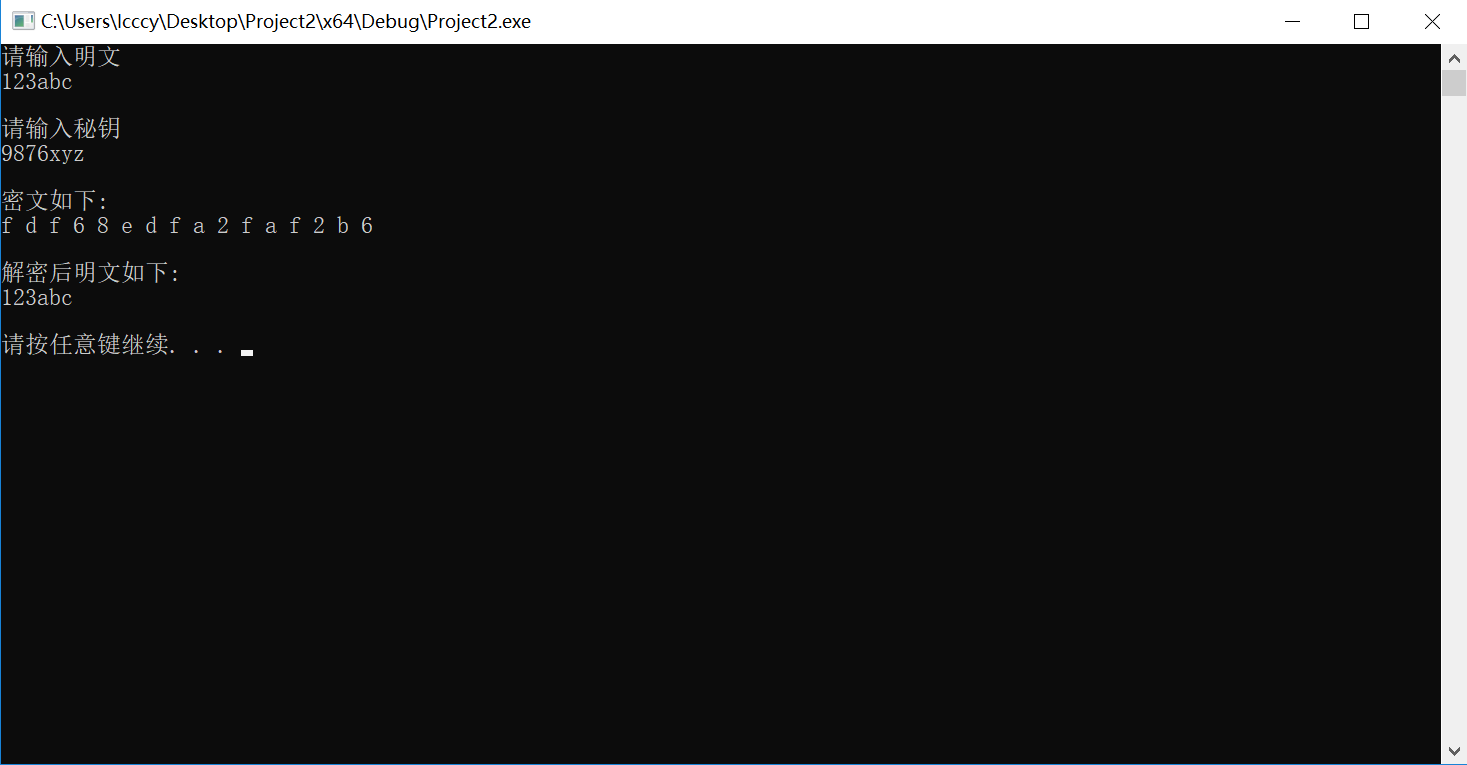


这样DES算法就算是完成了。

主程序如下：



编译运行结果如下：



总结：DES算法的设计过程老师的ppt上讲的算是挺清楚了，我们需要做的就是将这些理论转换为代码实现。通过这次作业，我对DES算法有了比较深入的了解，同时也提升了我程序设计的能力，受益匪浅。