测试结果

# 任务一：

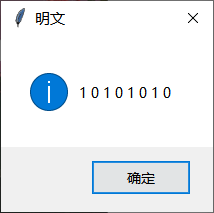
根据S-DES算法编写和调试程序，提供GUI解密支持用户交互。输入可以是8bit的数据和10bit的密钥，输出是8bit的密文。

调用encrypt中的的P\_Box()、F()函数可以对输入的明文进行加密，将输入为10bits的密钥进行加密并输出为长度为8bits的密文。

注：选择不同的功能时候应该注意也要输入对应格式的文本。

测试过程如下图所示：



# 任务二：

本算法已与多组同学（魏鹏坤组、方珩组、朱清扬组）进行共同测验，并检验通过。

# 任务三：

将输入的文本转化为对应的ASCⅡ编码下的字符，然后将每个字符转换为八位二进制数组，针对每个字符分别进行测试加解密最后呈现结果。

注：在这里我们发现，将ASCII码进行加解密时有可能会出现无法显示的情况，也就是说一些字符有可能会被加密为控制字符。因此ASCⅡ码的加密在部分情况下不能正常显示，所以最佳方案是将ASCⅡ进行“01”存储。

# 任务四：

假设你找到了使用相同密钥的明、密文对(一个或多个)，请尝试使用暴力破解的方法找到正确的密钥Key。以下展示几组不同明密文情况下暴力破解时得到的所有keys和破解时所需要的时间。破解的平均时间约为0.08s。

pla = [1, 1, 1, 1, 1, 1, 1, 1]

cip = [1, 1, 0, 0, 1, 1, 1, 0]

耗时为： 0.08201837539672852

密钥为 [1, 0, 1, 0, 1, 0, 0, 1, 1, 0]

密钥为 [1, 0, 1, 0, 1, 1, 0, 1, 1, 0]

密钥为 [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]

密钥为 [1, 1, 1, 0, 1, 1, 0, 1, 1, 0]

pla = [1, 0, 1, 0, 1, 0, 1, 0]

cip = [0, 0, 0, 1, 1, 0, 1, 1]

耗时为： 0.08490610122680664

密钥为 [1, 0, 1, 1, 1, 0, 0, 0, 0, 0]

密钥为 [1, 0, 1, 1, 1, 1, 0, 0, 0, 0]

密钥为 [1, 1, 1, 1, 1, 0, 0, 0, 0, 0]

密钥为 [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]

pla = [0, 0, 1, 1, 0, 0, 1, 0]

cip = [1, 0, 0, 0, 1, 0, 1, 1]

耗时为： 0.08501911163330078

密钥为 [0, 0, 1, 0, 0, 0, 1, 1, 1, 1]

密钥为 [0, 0, 1, 0, 0, 1, 1, 1, 1, 1]

密钥为 [0, 1, 1, 0, 0, 0, 1, 1, 1, 1]

密钥为 [0, 1, 1, 0, 0, 1, 1, 1, 1, 1]

# 任务五：

根据第4关的结果，进一步分析，对于你随机选择的一个明密文对，是否有不止一个key？进一步扩展，对应明文空间任意给定的明文分组，是否会出现选择不同的密钥加密得到相同密文的情况？

由第四关中的结果可知，对于同一组明密文存在不止一个key。

下面也对于同一明文选择不同的密钥加密能否得到相同密文的情况进行了举例说明：

对于明文: [1, 1, 1, 1, 0, 0, 1, 0]

密钥1 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] 和 密钥2 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0] 都可以加密出密文:

[0, 1, 0, 0, 0, 0, 1, 1]

对于明文: [1, 1, 1, 1, 0, 0, 1, 1]

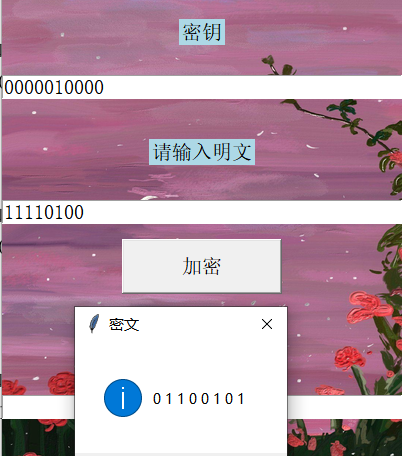
密钥1 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] 和 密钥2 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0] 都可以加密出密文:

[0, 1, 0, 0, 1, 0, 0, 1]

对于明文: [1, 1, 1, 1, 0, 1, 0, 0]

密钥1 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] 和 密钥2 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0] 都可以加密出密文:

[0, 1, 1, 0, 0, 1, 0, 1]

选择第三组结果在程序中进行测试，结果如下