# DES测试结果

**第一关：基本测试**



加密操作：

输入10bit密钥和8bit明文

密钥：1010101010

明文：10101010



获取加密结果



解密操作：

输入10bit密钥和8bit密文

密钥：1010101010

密文：10001111



获取解密结果



**第二关：交叉测试**

我们与其他小组的同学进行了二进制和ASCII的加密解密交叉测试，得出结果如下

* **二进制加密**

二进制明文和密钥相同时，加密所得密文也相同





* **二进制解密**

二进制密文和密钥相同时，解密所得明文也相同





* **ASCII加密**

ASCII明文和密钥相同时，加密所得密文也相同



****

* **ASCII解密**

ASCII密文和密钥相同时，解密所得明文也相同



****

**第三关：扩展功能**

对ASCII字符串进行加密

密钥：1010101010

明文：cat12345



对ASCII字符串进行解密

密文：sparking

密钥：1111111111



**第四关：暴力破解**

输入明密文对



开始破解



**第五关：封闭测试**

我们的暴力破解函数：

def find\_valid\_keys(self, pairs):  
 valid\_keys = [] # 用于存储有效密钥  
  
 for key\_candidate in range(1024): # 1024 = 2^10  
 key\_bits = self.sdes.int\_to\_bits(key\_candidate, 10) # 将密钥转换为10位二进制列表  
 matches = True # 假设密钥是有效的  
  
 # 检查所有的明密文对  
 for plaintext, ciphertext in pairs:  
 encrypted\_result = self.sdes.encrypt(plaintext, key\_bits)  
 # 如果任何一个加密结果不匹配，设置为 False  
 if encrypted\_result != ciphertext:  
 matches = False  
 break  
  
 if matches: # 如果所有明密文对都匹配  
 valid\_keys.append(key\_bits) # 添加到有效密钥列表  
  
 return valid\_keys

我们的破解函数能找到一对明密文之间的所有可能密钥，例如：



发现该明密文之间有四把不同的密钥，破解花费时间为0.0239秒