做题前首先熟悉python的使用，如struct.pack和struct.unpack，python和pyc文件反编译方面不多提，Life is short, you need python。

题目的难点主要在于对d函数的逆向，会有64次乘二，每次乘2后还会进行判断，如果大于0xffffffffffffffff，会&0xffffffffffffffff再^ 0xB0004B7679FA26B3。

我们在些逆向脚本时候同样要进行64次的除以二，但是有个问题就是如何处理&0xffffffffffffffff再^ 0xB0004B7679FA26B3呢？

其实能看出来的话这道题就很简单了，入手点就是奇偶。

每次乘二后这一次循环得到的是偶数，每次&0xffffffffffffffff再^ 0xB0004B7679FA26B3这一次循环得到的都是奇数。我们写脚本进行64次循环除以二时候，先进行判断，如果是偶数就直接除以二，如果是奇数就先^ 0xB0004B7679FA26B3，再+0xffffffffffffffff+1，再除以二即可。

题目源码：

# -\*- coding:utf-8 -\*-  
#Syc{L1fe\_i5\_sh0rt\_y0u\_n3ed\_py7h0n}  
import struct  
import time  
  
#控制64bit  
def b(a):  
 return a & 0xffffffffffffffff  
  
#将字符转换为小端序存储64bit的数值  
def c(str):  
 return struct.unpack("<Q",str)[0]  
  
#进行的主要运算  
def d(a):  
 for i in range(64):  
 a = a \* 2  
 if(a > 0xffffffffffffffff):  
 a = b(a)  
 a = b(a ^ 0xB0004B7679FA26B3)  
 return a  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 cmp\_data = [7966260180038414229L, 16286944838295011030L, 8598951912044448753L, 7047634009948092561L, 7308282357635670895L]  
  
 input = raw\_input("plz input your flag:")  
  
 #若不够8的倍数填充'\0'  
 if(len(input) % 8 != 0):  
 for i in range(8 - len(input) % 8):  
 input += '\00'  
  
 #每8个字节转换成一个数值进行运算  
 arr = []  
 for i in range(len(input) / 8):  
 value = d(c(input[i\*8:i\*8+8]))  
 arr.append(value)  
  
 #对比数据  
 for i in range(5):  
 if(arr[i] != cmp\_data[i]):  
 print "fail"  
 time.sleep(5)  
 exit()  
 print "success"  
 time.sleep(5)  
 exit()

脚本：

# -\*- coding:utf-8 -\*-  
import struct  
  
def decode(a):  
 for i in range(64):  
 #偶数  
 if(a % 2 == 0):  
 a /= 2  
 #奇数  
 else:  
 a ^= 0xB0004B7679FA26B3  
 a = a + 0xffffffffffffffff + 1  
 a /= 2  
 return a  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 cmp\_data = [7966260180038414229L, 16286944838295011030L, 8598951912044448753L, 7047634009948092561L, 7308282357635670895L]  
 flag = ""  
 for i in range(5):  
 flag += struct.pack(">Q",decode(cmp\_data[i]))[::-1].strip()  
 print flag

Flag：Syc{L1fe\_i5\_sh0rt\_y0u\_n3ed\_py7h0n}