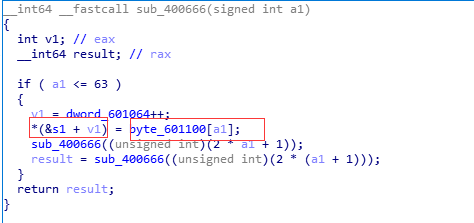
## 题目分析

首先程序会获取输入，然后对输入进行验证，这个验证是比较有意思的。





可以看到这是一个递归调用，sub\_400666函数中仍然调用了它本身这个函数，重要的地方是\*(&s1 + v1) = byte\_601100[a1];

&s1即是后面要比较的字符串

"bcec8d7dcda25d91ed3e0b720cbb6cf202b09fedbc3e017774273ef5d5581794"。

byte\_601100即是我们的输入，可以看到要比较的s1字符串是我们的输入进行打乱的结果。

也就是说只要找到flag每一位在比较字符串s1中对应的那一位就可以了。

然后我们怎么进行寻找对应的呢？我们可以写出同样的递归函数。

然后下面的比较是同样的逻辑，我们只需要使用一个就可以了。

## 脚本：

data = 0  
arr = []  
arr1 = []  
  
def fun(a):  
 global data  
 global arr  
 global arr1  
 if(a < 64):  
 arr.append(data)  
 arr1.append(a)  
 data = data + 1  
 fun(2\*a + 1)  
 fun(2\*(a + 1))  
  
def main():  
 fun(0)  
 str = "bcec8d7dcda25d91ed3e0b720cbb6cf202b09fedbc3e017774273ef5d5581794"  
 flag = [0 for x in range(64)]  
 for i in range(64):  
 flag[arr1[i]] = str[arr[i]]  
 print ''.join(flag)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()