**Python实验报告**

**实验题目：设计一个凯撒密码加密和解密程序**

**专 业：人工智能**

**姓 名：梁思越**

**学 号：22920192204002**

**实验日期：2021/3/25**

1. **实验内容**

设计实现针对中文的凯撒密码的加密和解密程序。

1. **实验步骤以及结果**

**1.加密部分**

**（1）数据描述：**

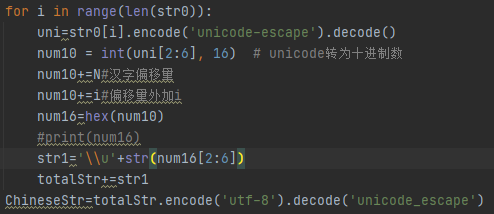
数据给定一个汉字的字符串，要求针对这一字符串进行凯撒密码加密。

**（2）方法描述：**

①设计输入函数来将明文保存在一个字符串中。

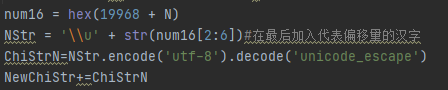
1.输入

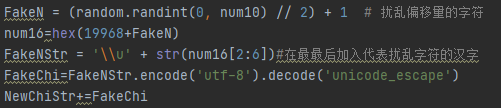
②利用随机数确定偏移量N。随机数求偏移量

③将每一个汉字转化为十进制的Unicode编码，再将Unicode编码作步长为N+i的移位操作，其中i为汉字在字符串中的位置。并再将新的Unicode编码转化为相应字符，组合而成的新字符串即为凯撒密码的加密模板。

④对新的字符串作步长为N的移位操作。字符串偏移N

⑤将偏移量N通过函数映射成Unicode编码，并将Unicode编码对应的字符加入新字符串中。



⑥用求偏移量N的方法再求出一个扰乱偏移量N的字符，并加入新字符串中，此时新字符串即为凯撒密码的暗文，加密完毕。

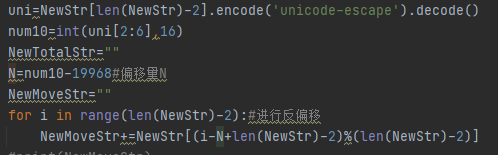
1. **解密部分**

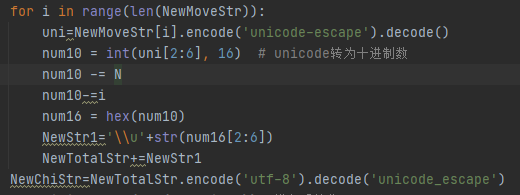
**（1）数据描述：**

数据给定一个字符串，要求针对这一字符串进行凯撒密码解密。

**（2）方法描述：**

①因为最后一位字符为扰乱偏移量的字符，所以无视。将倒数第二位字符转化为Unicode编码，并通过反函数映射出偏移量N，再对字符串作步长为-N的移位操作。

****

②将每一个汉字转化为十进制的Unicode编码，再将Unicode编码作步长为-(N+i)的移位操作，其中i为汉字在字符串中的位置。并再将新的Unicode编码转化为相应字符，组合而成的新字符串即为凯撒密码的明文，解密完毕。

1. **实验结果与分析**
2. **第一次加密操作**

**第一次加密**

1. **第一次解密操作**

**第一次解密**

1. **第二次加密操作**

**第二次加密**

**4.第二次解密操作**

**第二次解密**

1. **实验总结**

在这一次的实验中，当我最开始了解到实验内容后一度表示惊讶，认为以目前的Python基础是很难完成这一次实验的。但是，在一个下午的努力下，我还是靠自己编写出了一套属于自己的凯撒密码。

从这次的实验中，我了解到了许多Python的新知识，诸如随机数函数、字符与Unicode编码的相互转化等等。我感到受益匪浅，这对我的Python课程学习以及未来使用Python定下了良好的基础。

1. 附件：源码
2. 加密函数

import random  
def C2U2U2K():#中文转换unicode  
 str0=input("请输入明文")  
 #matrix=[[0]\*100]\*100#十六进制的unicode数组  
 uni=str0[len(str0)-1].encode('unicode-escape').decode()#第一个汉字转为unicode  
 num10=int(uni[2:6],16)#unicode转为十进制数  
 #N=num10%len(str0)+1#取模求偏移量  
 N = (random.randint(0,num10)//2) + 1 #随机数取模求偏移量  
 totalStr=""  
 for i in range(len(str0)):  
 uni=str0[i].encode('unicode-escape').decode()  
 num10 = int(uni[2:6], 16) # unicode转为十进制数  
 num10+=N#汉字偏移量  
 num10+=i#偏移量外加i  
 num16=hex(num10)  
 #print(num16)  
 str1='\\u'+str(num16[2:6])  
 totalStr+=str1  
 ChineseStr=totalStr.encode('utf-8').decode('unicode\_escape')  
 #print(ChineseStr)  
 NewChiStr=""  
 for i in range(len(ChineseStr)):#进行偏移  
 NewChiStr+=ChineseStr[(i+N)%len(ChineseStr)]  
 num16 = hex(19968 + N)  
 NStr = '\\u' + str(num16[2:6])#在最后加入代表偏移量的汉字  
 ChiStrN=NStr.encode('utf-8').decode('unicode\_escape')  
 NewChiStr+=ChiStrN  
 FakeN = (random.randint(0, num10) // 2) + 1 # 扰乱偏移量的字符  
 num16=hex(19968+FakeN)  
 FakeNStr = '\\u' + str(num16[2:6])#在最最后加入代表扰乱字符的汉字  
 FakeChi=FakeNStr.encode('utf-8').decode('unicode\_escape')  
 NewChiStr+=FakeChi  
 print("暗文为"+NewChiStr)  
 return NewChiStr

1. 解密函数

def K2C(NewStr):  
 uni=NewStr[len(NewStr)-2].encode('unicode-escape').decode()  
 num10=int(uni[2:6],16)  
 NewTotalStr=""  
 N=num10-19968#偏移量N  
 NewMoveStr=""  
 for i in range(len(NewStr)-2):#进行反偏移  
 NewMoveStr+=NewStr[(i-N+len(NewStr)-2)%(len(NewStr)-2)]  
 #print(NewMoveStr)  
 for i in range(len(NewMoveStr)):  
 uni=NewMoveStr[i].encode('unicode-escape').decode()  
 num10 = int(uni[2:6], 16) # unicode转为十进制数  
 num10 -= N  
 num10-=i  
 num16 = hex(num10)  
 NewStr1='\\u'+str(num16[2:6])  
 NewTotalStr+=NewStr1  
 NewChiStr=NewTotalStr.encode('utf-8').decode('unicode\_escape')  
 #for i in range(len(NewChiStr)):#进行反偏移  
 #NewMoveStr+=NewChiStr[(i-N+len(NewChiStr))%len(NewChiStr)]  
 print("明文为"+NewChiStr)