**Python 实验一**

**实验报告**

**实验题目：使用凯撒密码对信息进行加密及解密**

**专 业：人工智能专业**

**姓 名：林纪柏**

**学 号：22920192204006**

**实验日期: 2021.3.25**

1. **实验目的**

使用python对信息进行加密和解密，了解凯撒密码对中文字符串的特殊处理。初步认识python的基本结构与对各种函数的巧妙运用。

1. **实验内容**
2. 了解凯撒密码对英文字母加密的基本原理，思考其如何运用到中文加密中。
3. 加密程序只需要用户输入即可，不需要输入位移量n，n为程序随机序设置的量，输出其密文。
4. 解密程序只要求输入密文，输出正确信息。
5. **实验步骤**
6. 设置随机位移量n

使用random模块中的randint（）函数，程序即可产生随机数。

1. 信息的偏移处理

使用ord（）函数可以将字符转换成ACSII值，加上偏移量n后再通过chr（）函数将ACSII值转换成字符。

1. 偏移量n的传输

基本思想是将n隐藏于密码中，可用的方法是将n进行处理f（n），将密码的首字母加上f（n）的偏移量后的字符放于密码末。

1. 密码的设置

首先对f（n）对5取模得到e，再将信息按字符进行偏移n位，得到一位正确的密码，然后在每位正确的密码的后面设置e位干扰码，最后一位密码如步骤3所设。

1. 解密的方法

首先要得到偏移量n，即将输入的字符串末尾字符与首个的ACSII值相减得到f（n），再通过自定的逆f（）求得n以及f（n）%5得到步长e，之后按步长e得到正确的密码的位置，并使用ord（）与chr（）函数得到最终的信息。

1. **实验结果及展示**
2. 代码展示

加密程序：

inf=input("请输入你的信息:")  
import random  
key=random.randint(1,100) #字符偏移量  
code=list(inf) #字符串转列表

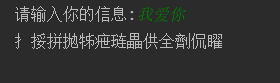
key2=key\*\*2+2016 #对key进行包装  
e=key2%5+1 #干扰码的个数  
i=0  
res="" #最终密码  
while i < len(code):  
 res+=chr(ord(code[i])+key) #信息字符偏移  
 j=0  
 while j<e: #添加干扰码  
 res+=chr(ord(code[i])+random.randint(1,1000))  
 j=j+1  
 i=i+1  
res+=chr(ord(res[0])+key2) #在密码末尾传入有关key2的信息  
print(res)

解密程序：

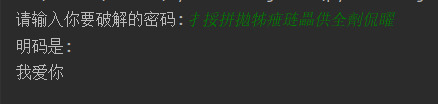
code=input("请输入你要破解的密码:")  
temp=list(code) #字符串转列表  
lenb=len(temp)  
import math  
key2=ord(temp[lenb-1])-ord(temp[0]) #列表首末字符的ASCII值相减得到key2  
key=int(math.sqrt(key2-2016)) #对key2解包装得到key  
e=key2%5+1 #由key2得到干扰码得个数  
  
inf="" #最终得信息  
for i in range(0,lenb-1,e+1): #跳过干扰码进行解密  
 inf+=chr(ord(temp[i])-key)  
print("明码是:")  
print(inf)

1. 样例展示
2. 信息：我爱你。

得到得密码是：



解码程序得到得明码是：

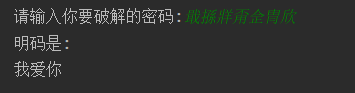


1. 信息：我爱你。

得到的密码是：

~~{H)3J_LWJYLK`88}7BS(0

解码得到的明码是：



1. **实验总结**
2. 本实验原本难度对于初学者来说是挺高的，在了解chr（）和ord（）两个函数后，难度便大大降低了，借此，我也认识到，python的强大之处不仅在于其易读性，更在于其函数的繁多且便利性。
3. 实验的另外一个难点在于如何设置密码才能提高复杂性，一开始只懂得简单地隐藏关键码，即偏移量，之后才了解了通过对关键码的包装和干扰码的设置能大大提高密码的复杂度，并且提高破解的难度，很大程度上杜绝了能通过大量数据来得到加密的原理的方法。