**Python程序设计实验一**

**实验报告**

**实验题目：设计一个凯撒密码（Caesar cipher）加密和解密程序**

**专 业：人工智能**

**姓 名： 黄祖荣**

**学 号： 22920192203974**

**实验日期**：**3月25日**

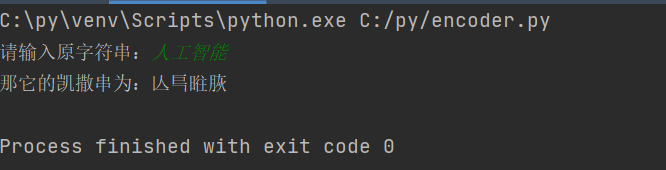
1. **实验目的**
2. **通过编写一定功能的程序以提高学生对python语言使用的熟练程度。**
3. **借助程序设计任务的要求与内容，来加强学生对python中一些语法结构，函数功能的理解与记忆。**
4. **提高学生在python程序编写过程中的调试以及debug能力。**
5. **帮助学生巩固在课堂上学习的python语法知识。**
6. **实验内容**

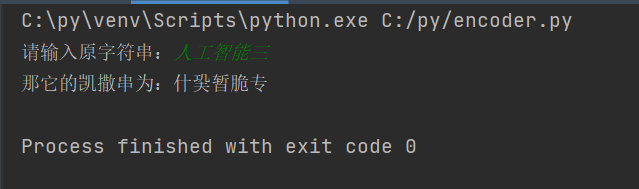
**用python程序语言设计两个程序，分别实现凯撒密码的加密和解密功能，要求凯撒密码的偏移量应当多样化，能达到较高的破译难度，并且具有一定的随机性。**

1. **实验步骤以及结果**
2. **首先，关于程序最基本的输入输出，我采取了先输入字符串plain，再将其转存到数组str当中，在str的存储过程中，逐位地对字符串中的字符加上其对应的asc码值偏移量的方法来达到字符串移位形成凯撒密码的效果，最终再用cipher=’’.join()这一方式将数组转化为字符串并输出。**
3. **本程序的核心部分应当在于偏移量的具体选取方式与保存方式，关于这个部分，我思索了许久，最终决定每一位字符的偏移量n由整个字符串中asc码值奇数或偶数的个数countodd或counteven，与每个字符在整个字符序列中的次序i所决定。**
4. **在设置密码时总共分为两步。首先，基于整个字符串长度len的奇偶性，先给字符串中每一个字符加上countodd\*2或counteven\*2的偏移量，这样就可以维持初步得到的子串每一位的奇偶性维持不变，可以维持编码、解码的一致性。例如，针对“人工智能”这一字符串，其总长度len为偶数，于是对其检索其中asc码值为偶数的字符个数，计算得为2，即counteven=2，那么则对每一位字符都有counteven\*2=4的位移量。**
5. **第二步，再对基于第一步得到的子串中每一个字符的asc码值加上其在整个字符串中对应的次序i。例如，对“人工智能”这一字符串，‘人’对应的第二步位移量i为0，但对于‘工’，其位移量则为1，‘智’的位移量则为2，以此类推。**
6. **值得注意的是，在解码程序中，应当先减去i再计算countodd或counteven，否则则会影响子串中字符奇偶个数的计算。**
7. **实验结果与分析**

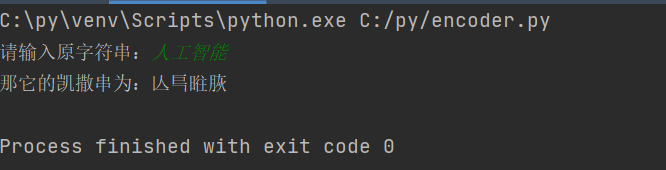
**基于上述的加密与解密规则，我们得到的cipher会因各种可能而表现出不同的区别，提高密码的破译难度，以下列举了部分情形，包括但不仅限于以下情形。**

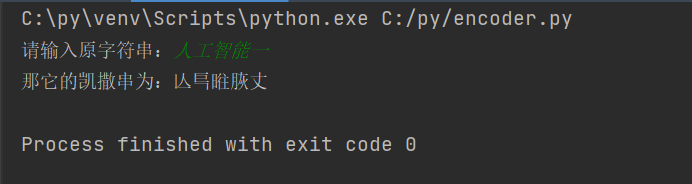
**1.有时因为多了一位字符，导致奇偶性发生变化，整体字符串偏移量出现了较大变化。**

****

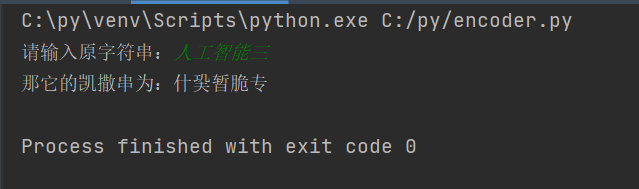
****

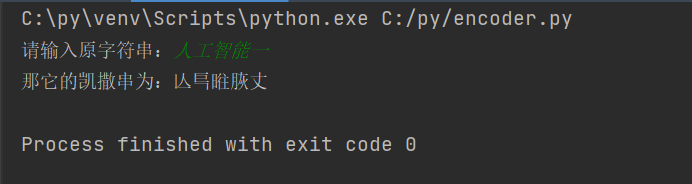
**但有时可能多加一位，也没有明显区别**

****

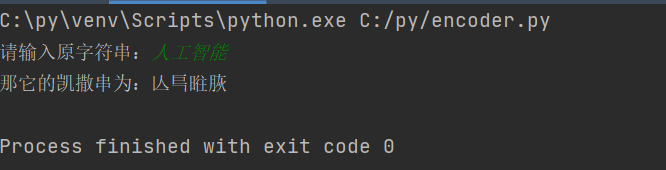
****

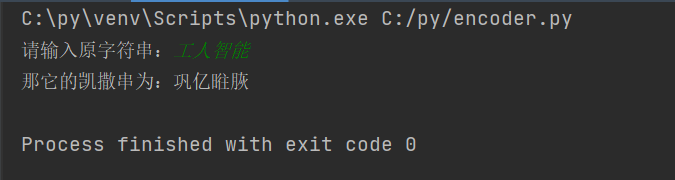
**2.仅仅因为一位字符的不同，导致了整个字符串相差甚远。**

****

****

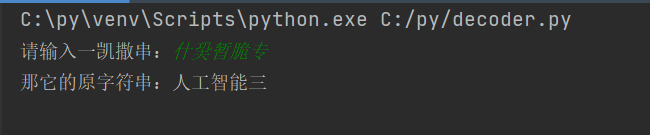
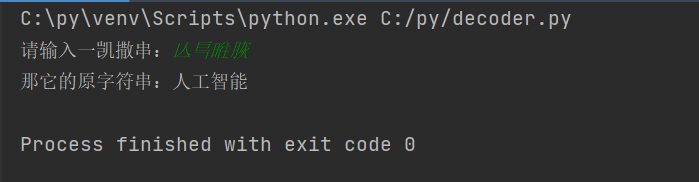
**3.基于偏移量的第二种成分，有按位的机制，交换其中两个字符，但表现出来的形式也差了很多，难以破译**

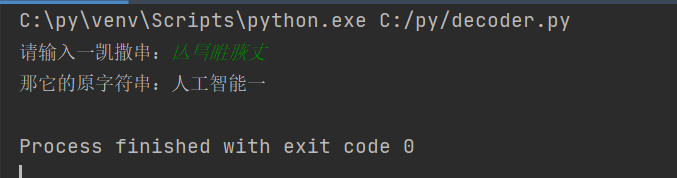
****

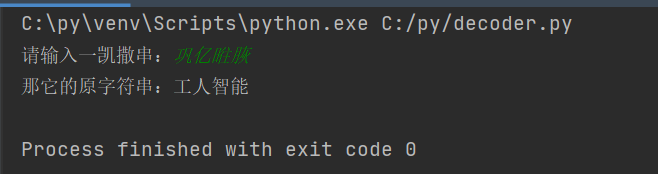
****

**以上仅列举了部分用例，还有别的可能出现的变化情况。除此之外，当字符串长度变长时，各种情况可能相互组合，产生更为复杂的情况，使得整个程序更难破译。**

**附各个测试用例的测试结果与程序实现的具体代码。**

****

****

****

**Encoder（加密程序）：**

def keywords(str):  
 sum = 0  
 i = 0  
 if len(str) % 2 == 1:  
 while i < len(str):  
 if ord(str[i]) % 2 == 1:  
 sum += 2  
 i += 1  
 return sum  
 else:  
 while i < len(str):  
 if ord(str[i]) % 2 == 0:  
 sum += 2  
 i += 1  
 return sum  
  
  
print("请输入原字符串：",end='')  
plain=input()  
str1=[]  
i=0  
while i<len(plain):  
 str1.append(chr(ord(plain[i])))  
 i+=1  
k=keywords(str1)  
i=0  
while i<len(plain):  
 str1[i]=chr(ord(str1[i])+k+i)  
 i+=1  
cipher=''.join(str1)  
print("那它的凯撒串为：",cipher,sep='')

**Decoder（解密器）：**

def keywords(str):  
 sum = 0  
 i = 0  
 if len(str) % 2 == 1:  
 while i < len(str):  
 if ord(str[i]) % 2 == 1:  
 sum += 2  
 i += 1  
 return sum  
 else:  
 while i < len(str):  
 if ord(str[i]) % 2 == 0:  
 sum += 2  
 i += 1  
 return sum  
  
print("请输入一凯撒串：",end='')  
cipher=input()  
str2=[]  
i=0  
while i<len(cipher):  
 str2.append(chr(ord(cipher[i])-i))  
 i+=1  
k=keywords(str2)  
i=0  
while i<len(cipher):  
 str2[i]=chr(ord(str2[i])-k)  
 i+=1  
plain=''.join(str2)  
print("那它的原字符串：",plain,sep='')

**五．实验总结**

**1.通过本次实验，熟悉了Python语言编程的基本方法，对各个数据类型有了一定的认识，并能借助函数实现数据类型之间的转换。**

**2.对凯撒密码的原理有了深刻的认识，并能够根据情况，设计简单的进阶凯撒密码，并通过程序实现加密，解密功能。**

**3.切身参与了python编程实验，能熟练掌握、应用从课堂与课本学习的语法知识，提高了python编程能力。**