**Python程序设计**

**实验报告**



**实验题目：**凯撒密码加密解密的中文实现

**专 业：**人工智能

**姓 名：**佟浡

**学 号：**22920192204166

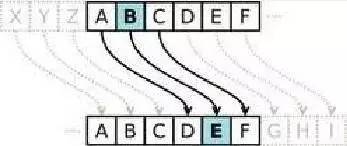
**实验日期：**2021.3.25

**诚信承诺：**我保证本次实验中的程序与报告内容均为本人自己编写

1. **实验目的**
2. 熟悉Python的开发环境，练习Python的运行技能
3. 通过编写程序，练习各种数据类型的应用，练习Python语言中的基本控制语句，熟练掌握缩进和注释等
4. 通过编写程序实现凯撒密码中文版的加密和解密过程，通过对加密方式的灵活设计，锻炼思维能力，提高编程素养。
5. **实验内容**

设计一个凯撒密码（Caesar cipher）加密和解密程序：

1. 一般的凯撒密码：给定移位长度N，按正常字母序列进行移位



1. 本次实验要求的凯撒密码：字母不再是常规的英文单词，而是汉字。且偏移量N应当随机生成，字符加密时的偏移不应呈现简单的线性关系。
2. **实验设计**
3. 汉字解决方式：通过ord函数提取出汉字的ASCII码，通过对ASCII直接进行值为n的偏移，然后把偏移之后得到的数值通过chr函数转化为字符，进而实现加密。
4. 偏移量n的设定：为了让破解者不易获得偏移量，每一个中文字符都设定了不同的偏移量，且这些偏移量均由random库中的randint函数随机生成。
5. 如何实现偏移量n在加密程序与解密程序中的传递：既要实现偏移量在两程序之间的传递，又要保证偏移量不被黑客截取，故我把随机生成的偏移量也通过chr函数转化成字符的形式，存储在加密后的字符当中，这样在生成加密文本的同时也间接告诉了使用者偏移量的值，但是这种方式又不便于黑客破解。
6. 程序源代码：

#加密

import random

a=input('请输入要进行加密的中文')

password= '' #定义一个空字符串存储加密后的文本

for i in a:

n = random.randint(1, 150) #随机生成偏移量

password = password + (chr(ord(i) + n))

#中文字符转ASCII码加偏移量再转换为字符，+表示两个字符串拼接

password = password + (chr(n))

#偏移量也转化为字符藏在加密字符中

print(password) #输出加密后的字符串

#解密

password=input('请输入要进行解密的中文')

password1='' #创建空字符串存储解密后的文本

j=0

for i in password:

n = (ord)(password[j+1])

#偏移量为被解密文字后面一个字符的ASCII码

password1=password1 + (chr((ord)(password[j]) - n))

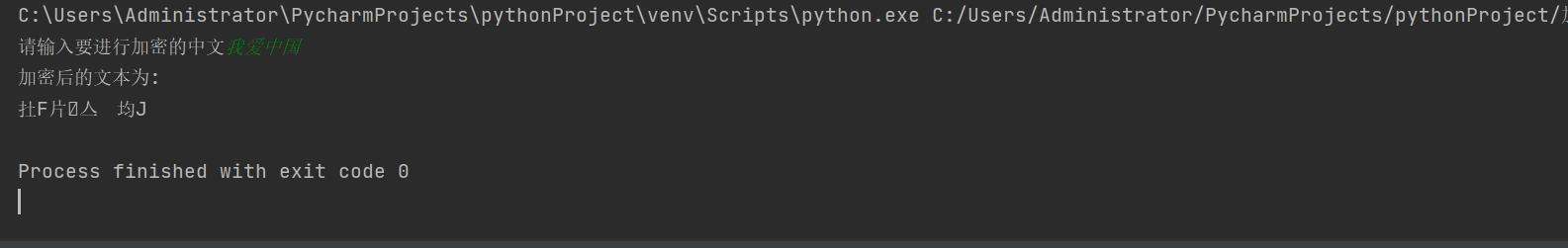
#被解密字符的ASCII码减去偏移量再转为字符型

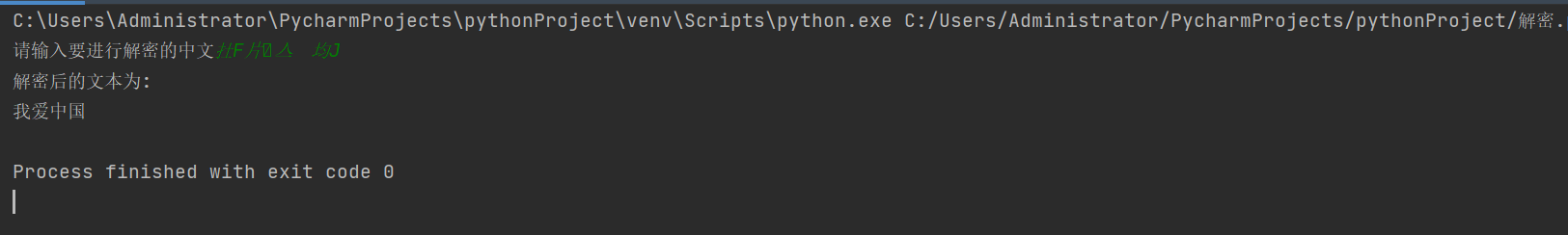
j=j+2

if(j>=len(password)): #循环终止条件

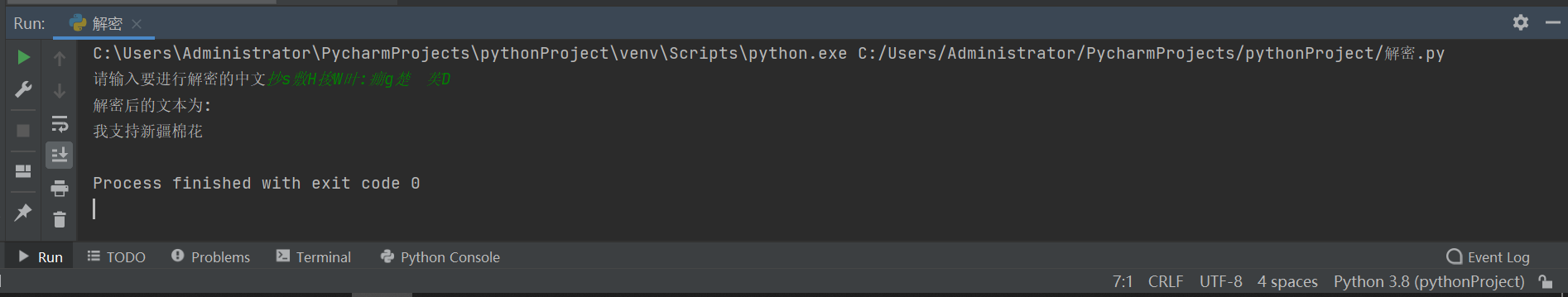
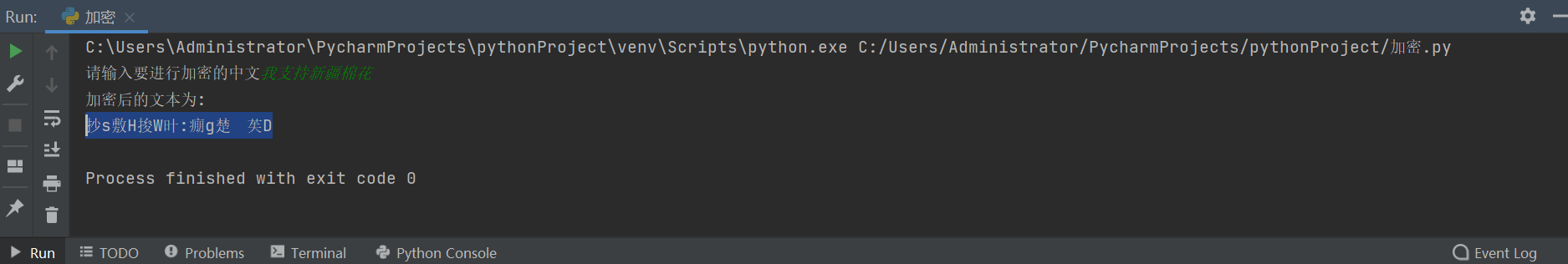
break

print(password1) #输出解密后的字符

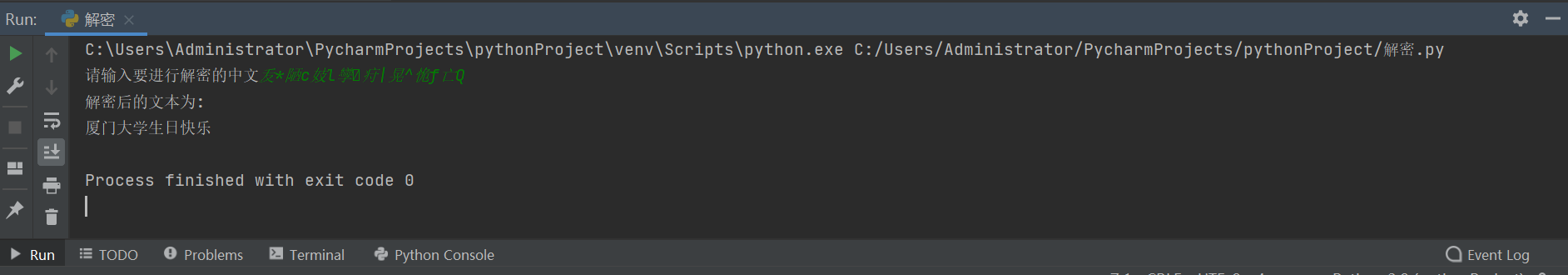
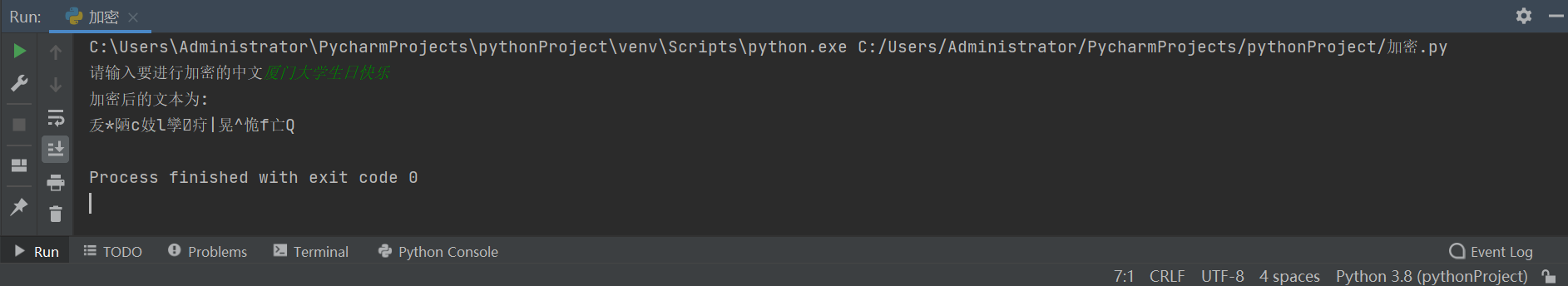
1. **实验结果与分析**
2. 测试样例一



1. 测试样例二



1. 测试样例三



1. 实验结论：

通过实验数据测试可以得到我设计的凯撒中文加密解密程序运行良好，确实能够达到预设的实验目的。

**五．实验总结**

1.语言学习层面：通过这次实验首先学会了python语句的基本书写方式，对于python各种数据类型有了更加深刻的了解，对于条件控制语句，循环控制语句有了更进一步的掌握。

2.程序设计层面：程序设计不仅要保证程序功能的实现，程序的安全性以及程序能否被正常使用也应当作为程序设计过程中应当被考虑的因素，在第一遍设计中，将加密程序与解密程序写在了同一个程序中，这显然不能满足程序被正确使用的基本条件。第二遍设计中，将偏移量设计成一种线性变化的量，但这种设计很容易被黑客破解，安全性不够。第三遍程序设计才同时满足上述两种条件，这样的程序才具有使用价值。