**Python实验报告**

**实验题目：凯撒密码**

**专 业：人工智能**

**姓 名：梁家铭**

**学 号：22920192204000**

**实验日期：2021.3.25**

**一. 实验目的**

学习运用python实现并改进凯撒密码

**二. 实验内容**

1.实现凯撒密码

2.针对中文改进

3.加入动态

**三．实验步骤以及结果**

算法思想：

将中文明文以utf-8编码，针对utf-8编码一个中文字符对应三个字节的特性，对每个字节加入两个随机生成的混淆字节以实现全部明文的动态变化，对于真正有效的字节再进行凯撒加密。

代码：

加密：

import random

def Caesar\_encode(plain\_text):

    plain\_text\_utf8 = list(plain\_text.encode("utf-8"))  # 每1个中文字符转成3个整数

    tmp = plain\_text\_utf8

    def func\_offset(num, i):

        if i % 3 == 0:

            base = 0xe5

            len = 0xe8 - 0xe5 + 1

        else:

            base = 0x80

            len = 0xbf - 0x80 + 1

        offset = 17

        return (num + offset - base) % len + base

    res = []

    for i, num in enumerate(tmp):

        num = func\_offset(num, i)

        if i % 3 == 0:

            res.append(num)

            res.append(random.randint(0x80, 0xbf))

            res.append(random.randint(0x80, 0xbf))

        elif i % 3 == 1:

            res.append(random.randint(0xe5, 0xe8))

            res.append(num)

            res.append(random.randint(0x80, 0xbf))

        else:

            res.append(random.randint(0xe5, 0xe8))

            res.append(random.randint(0x80, 0xbf))

            res.append(num)

    res = bytes(res)

    res = res.decode("utf-8")  # from byte to utf8

    return res

while True:

    text = input("加密输入1,退出输入其他\n")

    if text == '1':

        plain\_text = input("请输入明文\n")

        encrypted\_text = Caesar\_encode(plain\_text)

        print("密文是", encrypted\_text)

    else:

        break

解密：

def Caesar\_decode(text):

    tmp = list(text.encode("utf-8"))

    res = []

    def func\_offset(num, i):

        if i % 9 == 0:

            base = 0xe5

            len = 0xe8 - 0xe5 + 1

        else:

            base = 0x80

            len = 0xbf - 0x80 + 1

        offset = 17

        return (num - offset - base) % len + base

    for i, num in enumerate(tmp):

        if i % 9 == 0 or i % 9 == 4 or i % 9 == 8:

            res.append(func\_offset(num, i))

    res = bytes(res)

    res = res.decode("utf-8")

    return res

while True:

    text = input("解密输入1,退出输入其他\n")

    if text == '1':

        encrypted\_text = input("请输入密文\n")

        plain\_text = Caesar\_decode(encrypted\_text)

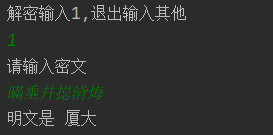
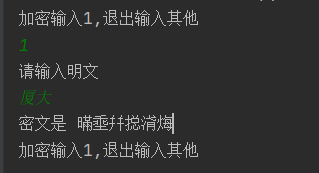
        print("明文是", plain\_text)

    else:

        break

**四、实验结果**

测试：

****

**五．实验总结**

1.虽然实现全明文的动态，但编码层仍是部分静态的，若知晓加解密方式可轻易破解。

2.偏移函数较为简单。

3.适用范围仅限于中文字符，若要适配更多字符需完善细节或更改编码方式。