Python算法之旅（第8期）

上期回顾：

描述：小王用计算机生成了10万个1000到1100之间随机整数，请对这些数据进行“去重”和“升序排序”的工作。

函数名：duplicate\_removal\_sort(t):

参数表：t -- 元组，其元素值为1000到1100之间的随机整数。

返回值：一个列表，存储了对元组进行“去重”和“升序排序”后的元素。

示例：对于t=(1095, 1036, 1095, 1036, 1026, 1036)，返回a=[1026, 1036, 1095]

参考代码如下：

def duplicate\_removal\_sort(t):

#把桶清空，因为数据范围是[1000,1100]，故只需101个桶

b = [0] \* 101

for num in t: #把数据装入对应桶中

b[num-1000] += 1

a = []

for i in range(101): #依次把数据从桶里倒出来

if b[i] > 0:

a.append(i+1000)

return a

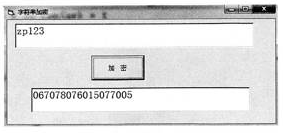
有关字符串加密和解密的问题越来越多了，2018年11月的高考卷17题算法颇为巧妙，接下来各地的模拟题纷纷仿效，力图吃透它，昨天刚考的宁波市高三期末“十校联考”卷第17题就是一列。

题目：字符串加密(2018 学年第一学期宁波市高三期末“十校联考”17题)

难度：4星 有趣：2星 有用：3星

分类：模拟，字符串

描述：在文本框Textl中输入待加密的n个字符(仅由ASCII码字符构成，最多支持960个字符)，输入后单击加密按钮，在文本框Text2中产生密文。加密方式如下：

①定义一个数组a(1 to 961)。 产生一个3到6之间的随机整数k, 将十进制数960均分成k份，字符在字符串中的位置除以k的余数决定该字符存放在第几份数据中(余数为1保存在第一份数据中，余数为2保存在第二份数据...余数为0保存在第k份数据中)；

②用十进制数127减去每个字符的ASCII码值，得到的差作为该字符的密文，并保存在数组a中，同一段内的密文依次存放；

③将随机产生的数k加64后保存在数组元素a(961)中，并一起参与加密；

④将数组a中所有有密文值的数组元素从后往前依次存放到数组b中；

⑤将数组b中的每个密文用3位数字保存，不足3位的前面用0补足，然后依次连接保存在变量sc中；

⑥在文本框Text2中输出se。

例如：

①若现有待加密的字符串为“zp123"，产生的随机数k=3，则960分成3份，每份可存放320个值，分别为a(1)至a(320)，a(321)至a(640)，a(641) 至a (960)；

②由于Asc(“z”)=122，则127-122=5。 字符“z”在待加密字符串中的位置是1，除以k的 余数为1，因此数字“5”放在第一份数据的第一个位置，即a(1)=5； 同理可得，第一份数据为a(1)=5， a(2)=77；第二份数据为a(321)=15， a(322)=76；第三份数据为a(641)=78；

③将随机产生的k与十进制数64的和保存到a数组的最后一个值中，即a (961)=64+3=67；

④将数组a中所有有密文值的数组元素从后往前依次存放到数组b中，得到

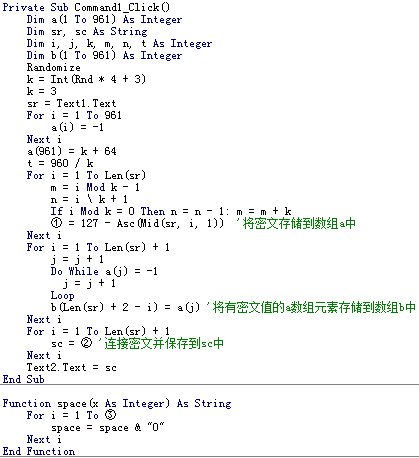
 b(1)=67，b(2)=78，b(3)=76， b(4)=15， b(5)=77，b(6)=5；

⑤将数组b中的每个元素用0补足3位后依次连接并保存在sc中，得到

sc = “067078076015077005”；

⑤输出sC。

注：（asc (“0”) =48，asc (“A”) =65，asc (“a”)=97）



(1)若加密后的密文为“068029041”，则随机数是\_\_\_\_\_\_ : 在TextI中输入的明文是\_\_\_\_\_\_\_

(2) )在空白处填写缺失的代码。

答案：（1） 4 (1分)；Vb (1分)

（2） a(m \* t + n) (1分)

sc & space(b(i)) & b(i) (2分)

4 - Len(Str(x)) (1分)

或

sc & Space(4 - Len(Str(b(i)))) & b(i)

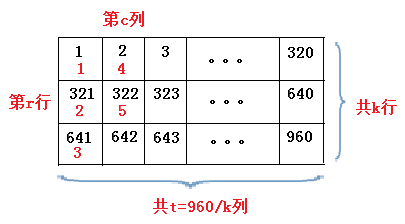
x

算法分析：

本题考查将数组均匀分段后再加密的算法，与2018年11月高考卷第17题有异曲同工之妙。此类题目的思考方法都是先构造矩阵，用二维数组的思维方式分析。因为高考VB中不能出现二维数组的表达式，所以最后再转换成一维数组来表达。

题目的意思是把960平分成k份，每份数据长度为t=960/k（k只能取3,4,5,6的原因是不能有余数），相当于构造一个k行，t列的矩阵。我们以题目提供的样例数据做示意图如下，黑色字体的1,2,3等数字表示数组a的下标，红色字体1,2,3,4,5代表字符串各字符的序号，例如对于字符串”zp123”，字符”p”序号为2，它的密文存储在a[321]中；字符”3”序号为5，它的密文存储在a[322]中。

第r行第c列的元素若用二维数组表示为a[r,c]，转换成一维数组就是a[(r-1)\*t+c]，题目提供的代码中与行号和列号有关的变量是m和n，其中m=r-1，n=c。



题目提供的代码有两处比较含蓄，需要耗费一定的时间才能理解。

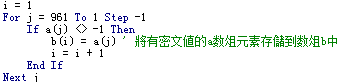
第一处是将密文存储到数组a时，若位置恰好在第k行，需要修正m和n的值，题目提供的代码是：If i Mod k = 0 Then n = n - 1: m = m + k

这个需要结合上面的代码来理解，比较耗费时间，更直观的代码是：

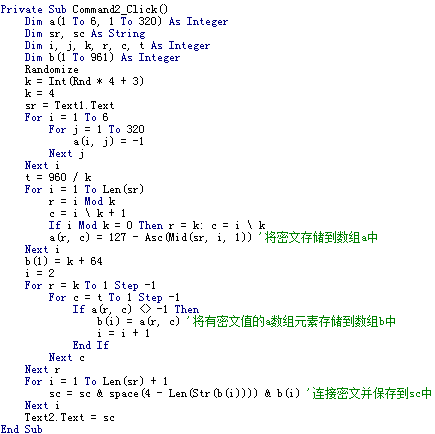
If i Mod k = 0 Then m = k - 1: n = i \ k

直接告诉读者数据在第k行，第i \ k列。

第二处是存储密文到数组b时，题目中的代码是将i和j的值递增，然后利用b(Len(sr) + 2 - i) = a(j)来实现逆序存储的功能，这里使用了一些技巧，造成了理解上的困难。更直观地方法是直接逆序扫描数组a，然后将密文存储到数组b，代码如下：



当然，如果能使用二维数组来表示，那代码就更简单直观了：



使用Python语言来解答此题，可以得到更简洁明了的代码。

描述：字符串加密。待加密的n个字符(仅由ASCII码字符构成，最多支持960个字符)，加密方式如下：

①产生一个3到6之间的随机整数k, 将十进制数960均分成k份，字符在字符串中的位置除以k的余数决定该字符存放在第几份数据中(余数为1保存在第一份数据中，余数为2保存在第二份数据...余数为0保存在第k份数据中)；

②用十进制数127减去每个字符的ASCII码值，得到的差作为该字符的密文，同一段内的密文依次存放；

③将随机产生的数k加64后作为第一个密文存放到数组b中；

④将其他所有有密文按照在数据段中的位置依次逆序存放到数组b中；

⑤将数组b中的每个密文用3位数字保存，不足3位的前面用0补足，然后依次连接成一个新的字符串；

⑥返回密文字符串。

函数名：encrypt (p)

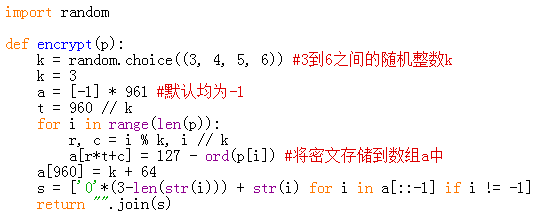
参数表：p –– 待加密的字符串。

返回值：s –– 加密后的字符串。

例1，当p = ”zp123”， k=3时，c=” 067078076015077005”；

例2，当p = ”Vb”， k=4时，c=” 068029041”。

代码如下：



课后思考：

前面我们根据字符串加密算法，获得了密文。现在的问题是：请你根据上述加密算法，逆向分析，设计一个解密程序，根据密文输出明文。

描述：加密算法如前述，设计一个解密程序，根据密文输出明文。

函数名：decrypt(s)

参数表：s –– 密文字符串。

返回值：p –– 明文字符串。

例1，当s=” 067078076015077005”时，p = ”zp123”；

例2，当s=” 068029041”时，p = ”Vb”。

另外，如果你有更 Pythonic（优雅的、地道的、整洁的）代码，或者与本文不同的算法思路和代码实现，请你一定留言或联系我，让我们一起讨论，共同进步。