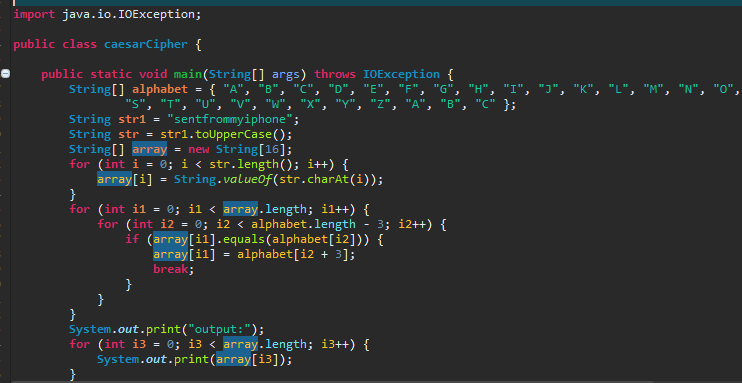
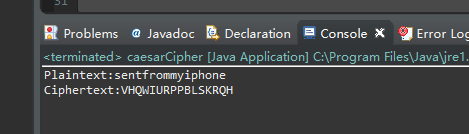
1. Caesar cipher

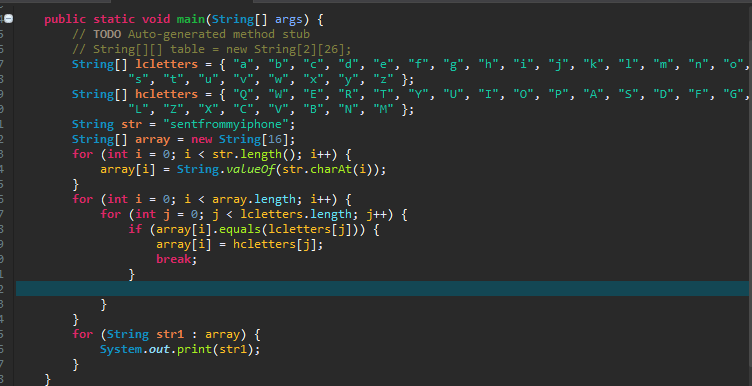


输出结果展示：

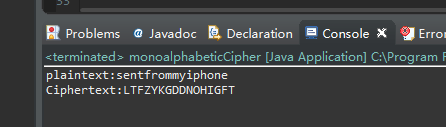


首先为所有字母创建一个String类型数组，创建一个字符串接收明文，并存在一个新数组中。通过for循环得到明文每一个字母在字母表中的位置并依次向后平移三位，为防止溢出。字母表在设计的时候增加了三位。

2.Monoalphabetic cipher:

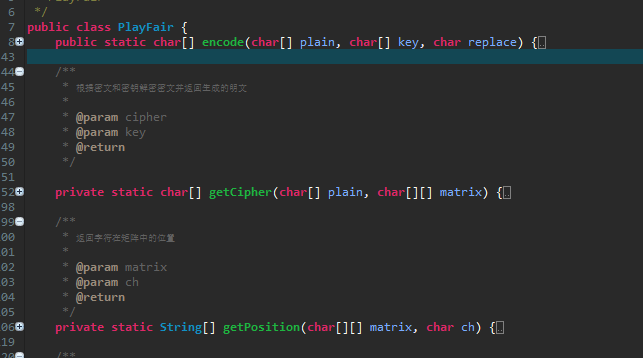


输出结果展示：



创建两个大小一致的string类型将key存储，并分别循环存储在两个不同的string类型数组中。通过明文与正常顺序的字母string数组进行比对，找到其位置信息，并将位置信息利用在第二个string类型数组中，将密文输出。

3.Playfair cipher：

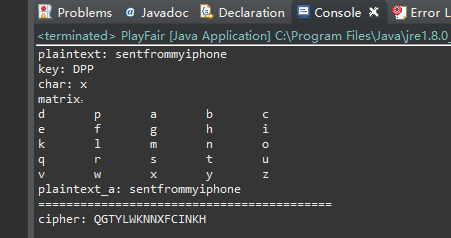


在类playfair中创建三个函数方法encode，getCipher，getPosition因为创建一个5×5的矩阵，所以这里将j,i看做同一个。getPosition中创建两种不同的pos,对拥有不同j,i的情况进行相同处理。在getCipher中通过行，列的比较，得到是i的处理还是j的处理。

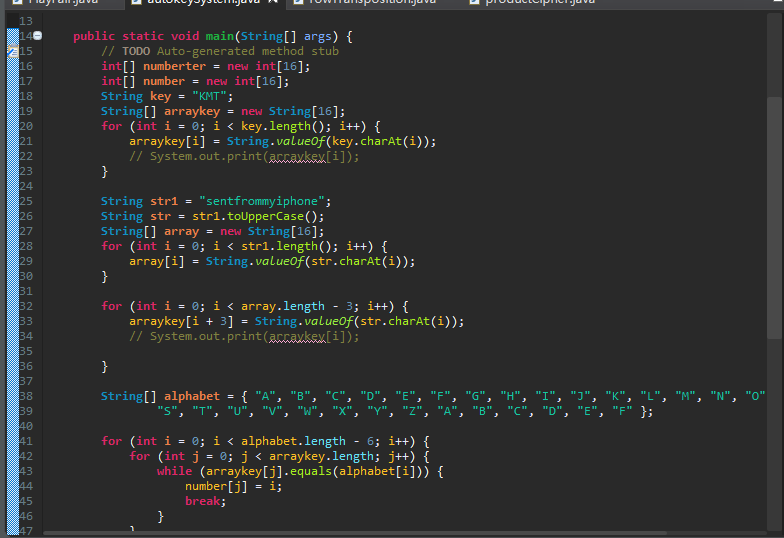


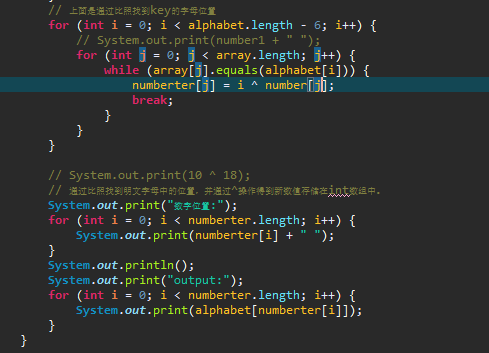
constructMatrix用以构造该矩阵。

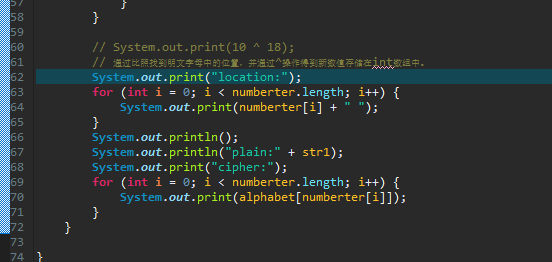
输出结果展示：



4.Vernam proposed the autokey system

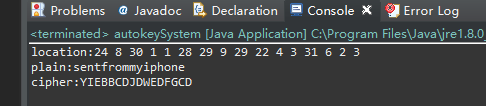




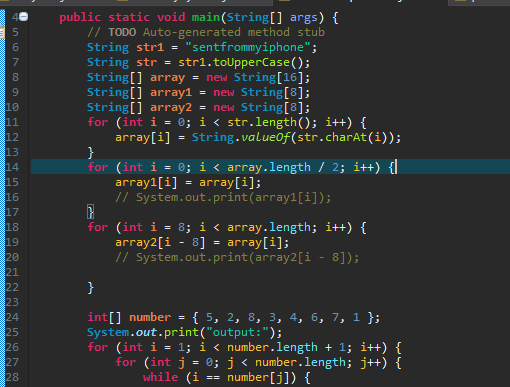


同样创建key的string类型数组，并且通过遍历得到每个元素字母的数字位置信息，同理遍历明文一样可以得到数字信息，将得到的两串数字异或操作得到新的一串数字，并将数字信息通过正常的字母顺序输出，为了防止溢出，原字母合集预留满足最大上限要求的空间。

部分结果展示：



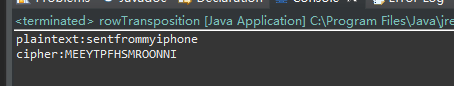
5.Row transposition:



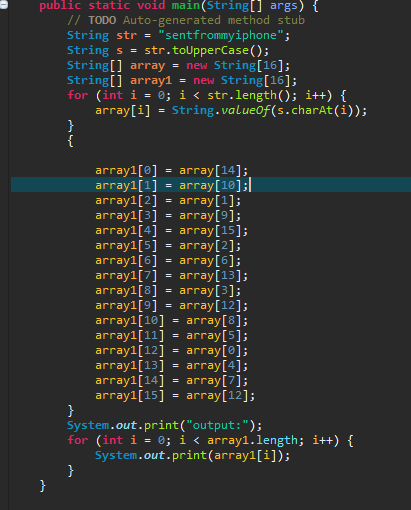
创建两个string类型数组，由于明文长度16，key长度8，所以需要两个。

按照key 52834671的数字顺序，将所在string数组依次输出。

部分结果展示：



6.Product cipher：



由于给出的key是按照一定顺序固定的排列且数量较少，可以依次为新string类型数组赋值。最后通过正常顺序一次输出即可。

部分结果展示：

