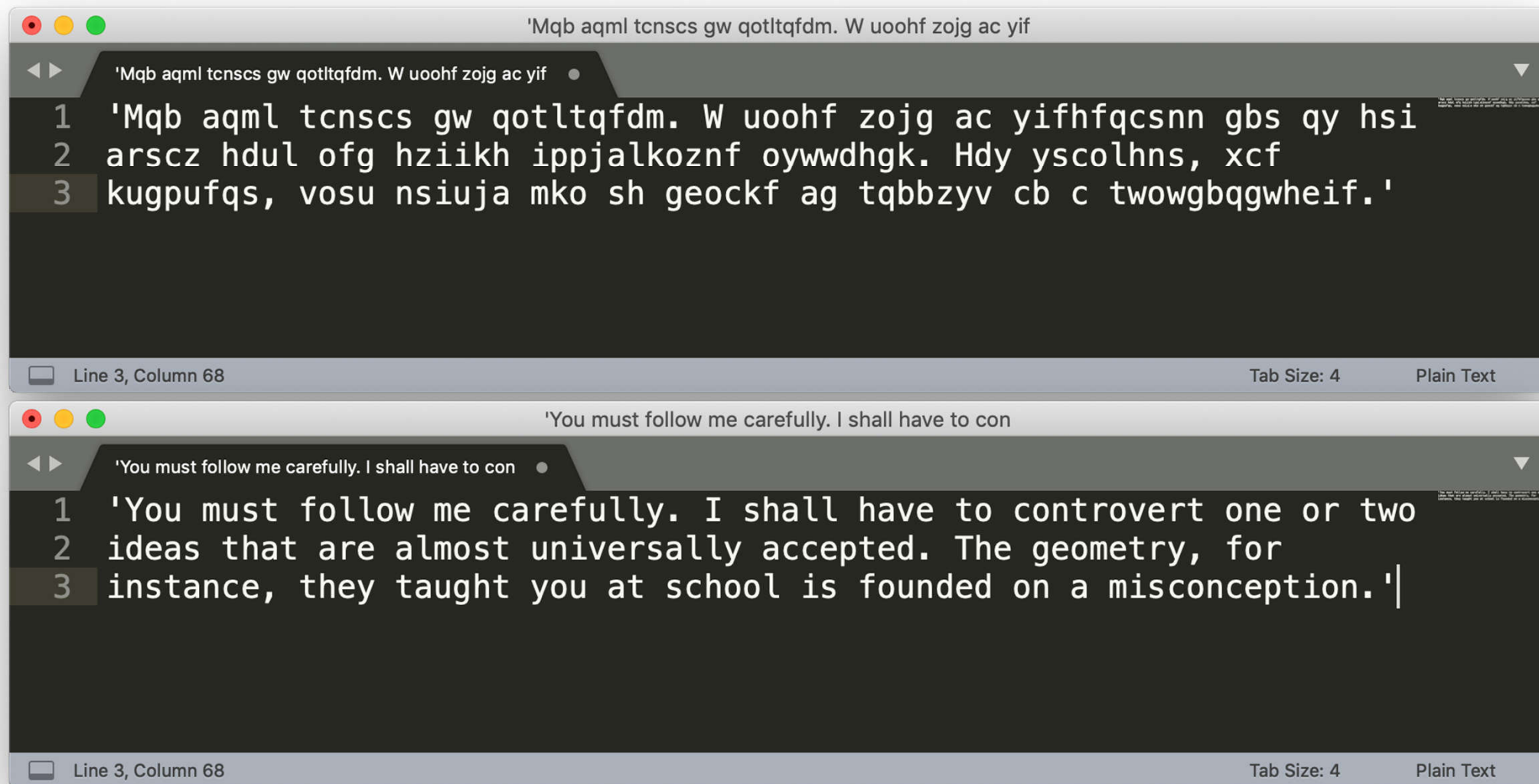


# Python程序设计

## 文件加解密

苏州大学计算机科学与技术学院

# 什么是加密



The image displays two screenshots of a code editor, likely VS Code, illustrating the concept of encryption. The top screenshot shows a file named 'Mqb aqml tcns cs gw qotltqfdm. W uoohf zojg ac yif'. The code content is encrypted and appears as a series of random characters. The bottom screenshot shows a file named 'You must follow me carefully. I shall have to con'. The code content is the same text as the top screenshot but is decrypted and readable. Both screenshots show the code in a dark-themed editor with line numbers 1, 2, and 3 visible on the left. The status bar at the bottom of each window indicates 'Line 3, Column 68', 'Tab Size: 4', and 'Plain Text'.

```
'Mqb aqml tcns cs gw qotltqfdm. W uoohf zojg ac yif
```

```
1 'Mqb aqml tcns cs gw qotltqfdm. W uoohf zojg ac yifhfqcsnn gbs qy hsi  
2 arscz hdul ofg hziikh ippjalkoznf oywwdhgk. Hdy yscolhns, xcf  
3 kugpufqs, vosu nsiuja mko sh geockf ag tqbbzyv cb c twowgbqgwheif.'
```

```
'You must follow me carefully. I shall have to con
```

```
1 'You must follow me carefully. I shall have to controvert one or two  
2 ideas that are almost universally accepted. The geometry, for  
3 instance, they taught you at school is founded on a misconception.'
```

# 维吉尼亚加密法

- 该方法最早记录在意大利密码学家吉奥万·巴蒂斯塔·贝拉索（Giovan Battista Bellaso）1553年的著作中
- 简单易用，初学者难以破译，被称为“不可破译的密码”
- 该加密算法的密钥是一系列字母，比如一个英文单词PIZZA
- 该单词密钥会分成若干个子密钥
  - 第1个子密钥是P，加密明文第1个字母
  - 第2个子密钥是I，加密明文第2个字母，其余如此类推
  - 加密明文第6个字母，则回过头来使用第1个子密钥P

# 维吉尼亚加密法

- 使用密钥**PIZZA**加密明文 ‘Common sense is not so common.’

明文字母	子密钥		密文字母
C (2)	P (15)	→	R (17)
o (14)	I (8)	→	w (22)
m (12)	Z (25)	→	l (11)
m (12)	Z (25)	→	l (11)
o (14)	A (0)	→	o (14)
n (13)	P (15)	→	c (2)
s (18)	I (8)	→	a (0)
e (4)	Z (25)	→	d (3)

- 密文是 ‘Rwlloc admst qr moi an bobunm.’

# 维吉尼亚加密法的实现

```
1  LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
2
3  def encrypt(msg, key):
4      return translate_msg(msg, key, 'encrypt')
5
6
7  def decrypt(msg, key):
8      return translate_msg(msg, key, 'decrypt')
```

- 字符串的find方法可以获取某个字母在该串中的索引

# 维吉尼亚加密法的实现

```
10 def translate_msg(msg, key, mode):
11     cipher = []
12     key_index = 0
13     key = key.upper()
14     for c in msg:
15         num = LETTERS.find(c.upper())
16         if num != -1:
17             if mode == 'encrypt':
18                 num += LETTERS.find(key[key_index])
19             elif mode == 'decrypt':
20                 num -= LETTERS.find(key[key_index])
21             num %= len(LETTERS)
22             if c.isupper():
23                 cipher.append(LETTERS[num])
24             else:
25                 cipher.append(LETTERS[num].lower())
26             key_index += 1
27             if key_index == len(key):
28                 key_index = 0
29         else:
30             cipher.append(c)
31     return ''.join(cipher)
```

# 测试维吉尼亚加密和解密函数

```
>>> msg = 'Common sense is not so common.'  
>>> key = 'PIZZA'  
>>> msg_en = encrypt(msg, key)  
>>> msg_en  
'Rwlloc admst qr moi an bobunm.'  
>>> msg_de = decrypt(msg_en, key)  
>>> msg_de  
'Common sense is not so common.'
```

# 加密文本文件

```
1  def encrypt_file(src_file, des_file, key):  
2      src_file_obj = open(src_file)  
3      msg = src_file_obj.read()  
4      src_file_obj.close()  
5      msg_en = encrypt(msg, key)  
6      des_file_obj = open(des_file, 'w')  
7      des_file_obj.write(msg_en)  
8      des_file_obj.close()
```



# 解密文本文件

```
10 def decrypt_file(src_file, des_file, key):  
11     src_file_obj = open(src_file)  
12     msg = src_file_obj.read()  
13     src_file_obj.close()  
14     msg_de = decrypt(msg, key)  
15     des_file_obj = open(des_file, 'w')  
16     des_file_obj.write(msg_de)  
17     des_file_obj.close()
```

# 思考题

- 在<https://inventwithpython.com/thetimemachine.txt>下载文本文件，使用维吉尼亚加密法对其加密，然后再对其解密，比较解密后的文件与原文件是否相同
- 破译维吉尼亚密码也不是特别困难的事情，一个基础步骤就是做频率分析，即统计字母在明文和密文里出现的频率。请编写一个程序，统计上述文本文件中英文字母的出现频率，按照频率从高到低的顺序输出，不区分大小写