

## Frequency Analysis

Enter the text you want to decode in the first line of the next cell. The output will be one big block of text that follows the uppercase convention for encrypted text."

---

```
1 cipherText = "GUVF VF GUR RAPELCGRQ GRKG LBH JNAG GB QRPBQR"
2 smashText = ""
3 for line in cipherText:
4     smashText = smashText + line.upper()
5     print smashText
6 cipherAlphabet = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S']
7
8 totalChars = float(smashText.count("")),
9 countList = [(letter,float(smashText.count(letter))/totalChars*100) for letter in cipherAlphabet]
10 sortedValues = sorted(countList, key=lambda freq: freq[1], reverse = True)
11 for i in sortedValues:
12     print i
```

---

The following is a frequency table for the english language: ....coming soon....

Based on this information, we might expect  $G$  and  $R$  to represent  $e$  and  $t$  in some order. Based on positions in the text, we guess that  $G \leftrightarrow t$  and  $R \leftrightarrow e$ .