Coding hints

For homework 2 (and some later homework assignments) you will probably want to use a computer and do a little coding. How much or how little of this you do is up to you. In theory, it's certainly possible to do everything by hand, but this will likely be quite tedious. (But hey, that's how people cracked Vigenère ciphertexts back in the 18th and 19th century, so just pretend you're Ulysses S. Grant!)

At the other extreme, if you're a skilled programmer, you could even automate the entire process. I recently wrote a function that goes through all the steps of cracking the Vigenère cipher, from start to finish; simply give it a ciphertext, and it spits out the most likely key and plaintext.

But you might find it easier to attack the problems in a more piecemeal fashion: write a bit of code here that helps with one step of the process, another bit of code there that helps with another step, etc. But then perhaps piece the key together by hand from information you got from each of those steps.

Once again, you are welcome to do this however you want, use whatever programming language you want, etc. You don't have to hand in any code. **What you turn in should include at least the correctly decrypted plaintext, the key, and ideally at least a little bit of other information about the steps you took to find the key.**

In class, I suggested Python as a very nice language to use for some of the coding we'll do in this class. If you happen to already know another high-level language such as Ruby or Lua, you might prefer to use one of those. But if you only know C or C++, and you'd like to use something easier for these assignments, my recommendation would be for Python.

Getting Python If your computer is a Mac, or if you have Linux installed, you can use Python simply by opening a terminal and typing "python". This will give you a simple (albeit somewhat rudimentary) way of interacting with Python. You just type in statements, and Python will run them on the fly and give you the results.

If you don't want to have to install software, you can use Python online with nothing but a web browser. There are many sites where you can try this for free:

- Wakari: https://www.wakari.io/
- Sage Math Cloud: https://cloud.sagemath.com/
- PythonAnywhere: https://www.pythonanywhere.com/

If you do want to install Python, I highly recommend using the IPython notebook, because it gives you a simple way to test out bits of code interactively, but still save the results.

Tips for using Python If you are new to Python, there is an excellent tutorial online at https://docs.python.org/3.5/tutorial/. Reading the first few chapters of that will give you plenty to get started with. (I'd recommend reading through chapter 5... but don't worry, chapters 1 through 3 are quite short, and chapter 2 can probably be skipped.)

Specifically for this assignment, here are a few really helpful tips:

- I defined functions called `tonum` and `tochar`. The first takes a capital letter and converts it to a number between 0 and 25. The second does the inverse. (These provide the *encoding* and *decoding*.) Feel free to copy these verbatim and use them:

```
def tonum(char):
    return ord(char) - 65

def tochar(num):
    return chr(num + 65)
```

- The `%` symbol gives the remainder when a number is divided by another:

```
41 % 26     # Returns 15
304 % 60    # Returns 4
-27 % 8     # Returns 5, because -27 = 8*(-4) + 5
```

- When you know the key length is k, use slicing to get every k-th letter of the ciphertext string:

```
text = "This is a test"
text[::4]   # Returns "T as", in other words, every 4th letter
text[2::4]  # Returns "ist", that is, every 4th letter starting with the 3rd
```

- Check out the `Counter` class in the `collections` module. It gives a stupidly easy way to count the number of occurrences of each letter in a chunk of text:

```
from collections import Counter
Counter("this is a test")
# Returns Counter({'t': 3, 's': 3, ' ': 3, 'i': 2, 'h': 1, 'a': 1, 'e': 1})
# Meaning "t" appeared 3 times, "h" appeared 1 time, etc.
```

- In Python, to iterate over two things at the same time in a `for` loop, use `zip`:

```
for ciphertextletter, keyletter in zip(ciphertext, key):
    # Do something with ciphertextletter and keyletter...
```

  For the above example, this is especially useful if you combine it with `cycle` from the itertools module: `from itertools import cycle`