22C:16 Programming Project 1
Grand Decryption Challenge

Due via ICON on Friday, April 8th, 4:59 pm

Problem. I will provide as input some text that has been encrypted by using a method that is a bit more sophisticated than Caeser's Cipher. Let me describe my encryption method first. Let π be a function that maps letters to letters; specifically π maps lower case letters to lower case letters and upper case letters to upper case letters. Assume that π is one-one, i.e., $\pi(x) \neq \pi(y)$ if $x \neq y$. Also assume that case is immaterial in the following sense, i.e., if $\pi(x) = y$ then $\pi(upper(x)) = \pi(upper(y))$. Here x and y are lower case letters and $upper(x)$ and $upper(y)$ are corresponding upper case letters. I will encrypt my text by replacing each letter x in the text by $\pi(x)$. These kinds of encryption schemes are called *substitution ciphers*.

Your goal is to write a program that reads the encrypted text and decrypts it. Of course, you don't know what π is and in a sense your program's goal is to figure out π.

More Details We will create some numbe of encrypted files and execute your programs using each of these as input. Your grade will be roughly proportional to the number of input files your program successfully decrypted. You can assume that all the encrypted files we will run your program on are created by encrypting "standard" English text files (e.g., articles we find online, parts of electronic versions of novels, etc.). Also, to make things simpler for you, we will leave all non-letter characters in the text (e.g., punctuation marks, white spaces, etc.) undisturbed.

Your program should start by prompting the user for an input file name and then an output file name. After that your program will work silently, reading from the specified input file and writing into the specified output file.

Techniques There is no way for the program to be sure that it is correctly decrypting the file and so it should try to be as confident of its decryption as it possibly can. You should feel free to use as many tricks as you want. Here are a few of my suggestions.

1. There are only a small number of valid single letter words and two-letter words in English. Paying attention to these will reduce the possibilities you have to consider.

2. If you had a dictionary of valid English words at your disposal, you could try and match partially decrypted words with words in your dictionary to further reduce the possibilities.

3. Another approach that should help you is called *frequency analysis*. Here is some text that I have copied from Wikipedia's article on frequency analysis:

   > Moreover, there is a characteristic distribution of letters that is roughly the same for almost all samples of that language. For instance, given a section of English language, E, T, A and O are the most common, while Z, Q and X are rare. Likewise, TH, ER, ON, and AN are the most common common pairs of letters (termed bigrams or digraphs), and SS , EE , TT , and FF are the most common repeats. The nonsense phrase "ETAOIN SHRDLU" represents the 12 most frequent letters in typical English language text.

   Paying attention to the frequencies of single letters, two-letter combinations, etc. should help significantly in deciphering the given text.

Final words Your biggest challenge will be to organize all of the things you could do into a few self-contained pieces that you can think about in isolation. After you get to this point, you can deleop algorithms for these pieces, translate these algorithms into code (using functions), and finally put this all together into a working program. The TAs and I will provide a lot of guidance on this and you have plenty of time to think about this and do an excellent job. Feel free to look up Wikipedia's article on substitution ciphers for more ideas.
