Project 1

EN.500.133 Bootcamp: Python, Intersession 2023

January 3, 2023

1 Problem A

Intelligence agents have intercepted encrypted messages between two officials of a hostile foreign government. An analyst in your organization has figured out that the message has been encrypted using this simple scheme: a is replaced by z, b by y, c by x and so on (see Figure 1). Specifically, "hello" would be encrypted as "svool".

Task: In a file named reverse.py, write a Python script to decipher any encrypted message given as input. You should assume input is always in lower-case. Punctuation and whitespace characters are not modified by the encryption scheme - in other words, these characters map to themselves. See the sample outputs below. For autograding purposes, please ensure that your framing text for input ('Enter the ...') and output ('The plaintext ...') match the samples below.

a	b	c	d	e	f	g	h	i	j	k	l	m
⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓
z	y	x	w	v	u	t	s	r	q	p	o	n

n	o	p	q	r	s	t	u	v	w	x	y	z
⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓
m	l	k	j	i	h	g	f	e	d	c	b	a

Figure 1: Scheme employed to encrypt the message. The arrows indicate how letters of the plaintext are mapped to the letters in the encrypted version of the message.

1.1 Sample Output 1

Enter the encrypted message: **kizxgrxv hlxrzo wrhgzmxrmt**
The plaintext message is: **practice social distancing**

1.2 Sample Output 2

Enter the encrypted message: **blf ziv zdvhlnv**
The plaintext message is: **you are awesome**

2 Problem B

Officials of the hostile foreign government suspect that you may have found a way to crack their code, so they have upgraded their encryption technology. Luckily, their new technology is also easy to break. The encryption strategy involves a simple shifting approach such as that as shown in Figure 2.

a	b	c	d	e	f	g	h	i	j	k	l	m
⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓
x	y	z	a	b	c	d	e	f	g	h	i	j

n	o	p	q	r	s	t	u	v	w	x	y	z
⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓
k	l	m	n	o	p	q	r	s	t	u	v	w

Figure 2: The key in this improved scheme is just an integer value indicating the shift amount used to perform the mapping to an alphabetical list of letters shifted rightward by a value between 1 and 25. In this diagram, the quantity `shifted_by` is 3.

There are 25 possible values for the exact value of `shifted_by` (i.e., assume that the plaintext is not equal to the ciphertext). Your co-worker does not know which value is in use. You propose to write code that decrypts the message based on all 25 keys. You suggest that the right message can be picked out manually after reading all 25 outputs. However due to the high volume of messages to decrypt, you have been asked to automate this process. For a given input they want your script to give one output, not 25.

Your co-worker suggests your program could decrypt the message all 25 possible ways and pick the one that most resembles typical English text. She suggests that you calculate a score named *chi* (χ) for each of the 25 possibilities where χ is defined as:

$$\chi_{key} = \sum_{\alpha} \frac{(n_\alpha - p_\alpha N)^2}{p_\alpha N}$$

Here the subscript α denotes the characters from a to z. The sum in the equation above has 26 terms. The value N is the length of the secret message you are trying to crack. For each character α, the value n_α is the number of times character α actually occurs in the decrypted message under the given key named key. Finally, p_α is the likelihood of an occurrence of character α in normal English text. The output with the lowest value χ is likely the secret message, so it is the one your program should output.

In order to find the likelihood values p_α, one can analyze many large pieces of English text and count the number of times x_α a certain character α occurs in them. If the body of text has a total of M characters overall, then $p_\alpha = x_\alpha/M$.

Since you have not yet learned how to use files in Python, below is a small piece of code that will help you analyze text stored in a file called sample.txt. The code below opens a file and reads it line by line and prints the output. You may modify and use this piece of code to calculate p_α. You may use the provided file, pride_prejudice.txt, a plaintext version of the Jane Austen classic, to accomplish the frequency analysis.

```python
# Open file titled sample.txt
my_sample_file = open('sample.txt', 'r', encoding='utf-8-sig')

# Iterate through each line in file my_sample_file
for line in my_sample_file:
    # Print each line
    print(line)

# Close the file
my_sample_file.close()
```

Task: In a file named shifted.py, write a script that implements the above plan. Again, for autograding purposes, please ensure that your framing text for input ('Enter the ...') and output ('The plaintext ...') match the samples below.

2.1 Sample Output Example

Enter the encrypted message: **zlnlshgld lv wkh ehvw wklqj hyhu dqbrqh lq wkh zruog fdq zulwh dqbwklqj wkhb zdqw derxw dqb vxemhfw vr brx nqrz brx duh jhwwlqj wkh ehvw srvvleoh lqirupdwlrq vdbv pb ervv plfkdho vfrww**
The plaintext message is: **wikipedia is the best thing ever anyone in the world can write anything they want about any subject so you know you are getting the best possible information says my boss michael scott**

3 Instructions for submission

- Submit the two files (i) reverse.py and (ii) shifted.py to Gradescope. Be sure your name and JHED appear in a comment line at the top of each file.
- When each submitted file is executed, it should each accept one input string and print the decrypted message. See the expected sample outputs shown above.
- Include substantial comments in your code. Your code will be evaluated not only on correctness, but also on style and good programming techniques.

- If your solution is not working perfectly, turn it in as-is with detailed comments describing which parts are complete, and which still need work. Be sure that what you turn in runs, to make it possible to receive feedback and partial credit.

- You are strongly encouraged to adopt an incremental coding style, making small changes to your code one at a time, so that you always have a version of your program which runs, meaning you will always have something to turn in, even if it is not 100% complete. You are also reminded to exercise discipline in backing up your work.If you have trouble or need extra help, don't hesitate to attend office hours or contact a member of the course staff through Piazza.

- You are permitted to discuss the instructions with other students for clarification. However, you may not discuss any substance of your solution with anyone except course staff, whether that be algorithms, test cases, component design, or actual code.