XOR decryption

Problem 59 (http://projecteuler.net/problem=059)

Each character on a computer is assigned a unique code and the preferred standard is ASCII (American Standard Code for Information Interchange). For example, uppercase A = 65, asterisk (*) = 42, and lowercase k = 107.

A modern encryption method is to take a text file, convert the bytes to ASCII, then XOR each byte with a given value, taken from a secret key. The advantage with the XOR function is that using the same encryption key on the cipher text, restores the plain text; for example, 65 XOR 42 = 107, then 107 XOR 42 = 65.

For unbreakable encryption, the key is the same length as the plain text message, and the key is made up of random bytes. The user would keep the encrypted message and the encryption key in different locations, and without both "halves", it is impossible to decrypt the message.

Unfortunately, this method is impractical for most users, so the modified method is to use a password as a key. If the password is shorter than the message, which is likely, the key is repeated cyclically throughout the message. The balance for this method is using a sufficiently long password key for security, but short enough to be memorable.

Your task has been made easy, as the encryption key consists of three lower case characters. Using p059_cipher.txt (https://github.com/amarovita/projecteuler/raw/master/p059_cipher.txt) (right click and 'Save Link/Target As...'), a file containing the encrypted ASCII codes, and the knowledge that the plain text must contain common English words, decrypt the message and find the sum of the ASCII values in the original text.

Solution

```python
from itertools import permutations
import string

with open('p059_cipher.txt', 'r') as f:
    data = list(map(int,f.read().split(',')))
keynum = (len(data) // 3) + 1

for pwd in permutations(list(map(ord,string.ascii_lowercase)), 3):
    pwd = pwd * keynum
    text = [a ^ b for a, b in zip(data, pwd)]
    utext = ''.join(map(chr,text)).upper()
    if ' A ' in utext and ' THE ' in utext and ' AN ' in utext:
        print('TEXT: ', ''.join(map(chr,text)), '\nKEY:', ''.join(map(chr,p
        break
```

TEXT: An extract taken from the introduction of one of Euler's most celeb
rated papers, "De summis serierum reciprocarum" [On the sums of series of
reciprocals]: I have recently found, quite unexpectedly, an elegant expres
sion for the entire sum of this series 1 + 1/4 + 1/9 + 1/16 + etc., which
depends on the quadrature of the circle, so that if the true sum of this s
eries is obtained, from it at once the quadrature of the circle follows. N
amely, I have found that the sum of this series is a sixth part of the squ
are of the perimeter of the circle whose diameter is 1; or by putting the
sum of this series equal to s, it has the ratio sqrt(6) multiplied by s to
1 of the perimeter to the diameter. I will soon show that the sum of this
series to be approximately 1.644934066842264364; and from multiplying this
number by six, and then taking the square root, the number
3.141592653589793238 is indeed produced, which expresses the perimeter of
a circle whose diameter is 1. Following again the same steps by which I ha
d arrived at this sum, I have discovered that the sum of the series 1 +
1/16 + 1/81 + 1/256 + 1/625 + etc. also depends on the quadrature of the c
ircle. Namely, the sum of this multiplied by 90 gives the biquadrate (four
th power) of the circumference of the perimeter of a circle whose diameter
is 1. And by similar reasoning I have likewise been able to determine the
sums of the subsequent series in which the exponents are even numbers.
KEY: exp
SUM: 129448