ASSIGNMENT-2

Team Name: codemeisters

Members:

VIJIT MALIK (170791)

HITESH KUMAR (170305)

PRAVEEN KUMAR (170504)

STEPS:

1. IDENTIFICATION OF THE TYPE OF ENCRYPTION.

We entered read in the command and came across the following cipher text.

Lg ccud qh urg tgay ejbwdkt, wmgtf su bgud nkudnk lrd vjfbg. Yrhfm qvd vng sfuuxytj "vkj_ecwo_ogp_ej_rnfkukf" wt iq urtuwjm. Ocz iqa jdag vio uzthsivi pqx vkj pgyd encpggt. Uy hopg yjg fhkz arz hkscv ckoq pgfn vu wwygt nkioe zttft djkth.

As professor has taught us in class about the various types of ciphers like substitution cipher, permutation cipher and so on. Our first guess was to use permutation cipher. As permutation cipher key is a multiple of the ciphertext length, I used the following code

```
plain_text = "Lg ccud qh urg tgay ejbwdkt, wmgtf su bgud nkudnk lrd vjfbg. Yrhfm qvd vng sfuuxytj vkj_ecwo_ogp_ej_rnfkukf wt iq urtuwjm. Ocz iqa jdag vio uzthsivi pqx vkj pgyd encpggt. Uy hopg yjg fhkz arz hkscv ckoq pgfn vu wwygt nkioe zttft djkth."
print(len(plain_text))

text = ""
for i in plain_text:
  if(i=='.'):
    continue
  if i==' ' :
    continue
  if i==',':
    continue
  text = text + i


print(text)
len(text)
```

The text length after removing inverted commas and underscores came out to be 185. It's multiples are 5 and 37. 37 being too big of a size for the key was ignored. I used 5 and looked at the first 5 letters and permuted them but could not find any English word.

Then we assumed it to be a substitution cipher and did it's frequency analysis and determined that was not substitution cipher (as it was not even close to the frequency of letters in reality). For referring to frequency analysis code you can check our previous assignment.

```
G   16× 8.65%
T   13× 7.03%
U   13× 7.03%
K   12× 6.49%
J   10× 5.41%
F   9×  4.86%
V   9×  4.86%
D   9×  4.86%
H   7×  3.78%
N   7×  3.78%
Y   7×  3.78%
O   7×  3.78%
W   7×  3.78%
C   7×  3.78%
R   6×  3.24%
Q   6×  3.24%
I   6×  3.24%
P   6×  3.24%
Z   5×  2.7%
E   5×  2.7%
S   4×  2.16%
A   4×  2.16%
B   3×  1.62%
M   3×  1.62%
X   2×  1.08%
L   2×  1.08%
```

When we were getting nowhere we just typed all the commands in the input due to frustration. To our surprise we came across this when we typed GO as input:

According to the instructions provided, we counted the number of characters in the horizontal segments and got this:

10-2-6-2-3-5-2-2-1

It looked like a key for decrypting the encrypted text, but it was not a permutation. Also as all values were less than 26, so we converted these values into alphabets we get:

Key1: “jbfbcebba” (conversion from index 1 corresponding to ‘a’)
Key2: “kcgcdfccb” (conversion from index 0 corresponding to ‘a’)

Assuming this was the key, we started surfing the internet about the ciphers with alphabetical keys and came across the Vigenere Cipher.

2. BREAKING THE ENCRYPTED TEXT:

Fortunately, our first try was upon the vigenere cipher and hence we were quick to break the coded message. We used the following code:

```
cipher_text = 'Lg ccud qh urg tgay ejbwdkt, wmgtf su bgud nkudnk lrd vjfbg. Yrhfm qvd vng sfuuxytj "vkj_ecwo_ogp_ej_rnfkukf" wt iq urtuwjm. Ocz iqa jdag vio uzthsivi pqx vkj pgyd encpggt. Uy hopg yjg fhkz arz hkscv ckoq pgfn vu wwygt nkioe zttft djkth.'

print(len(cipher_text))

key = "kcgcdfccb"
ex_key = ""
index = 0

while(len(ex_key)<235):
  ex_key = ex_key + key[index] ;
  index=index+1
  if(index==len(key)):
    index=0

index=0
plain_text = ""

for i in cipher_text:
  ascii_val = ord(i)

  if(ascii_val>=97 and ascii_val<=122):
        val = (ascii_val - ord(ex_key[index]) + 26)%26
        plain_text = plain_text + chr(val + 97)
        index = index + 1
        continue;

  if(ascii_val>=65 and ascii_val<=90):
        val = (ascii_val - ord(ex_key[index].upper()) + 26)%26
        plain_text = plain_text + chr(val + 65)
        index = index + 1
        continue;
```

```
    plain_text = plain_text + i

print(plain_text)
```

The plain text came out to be this for key1:

“Cf xbsz pg uif ofyu dibncfs, uifsf jt wfsz mjuumf kpz uifsf. Tqfbl pvu uif qbttxpse "uif_dbwf_nbo_cf_qmfbtfe" up hp uispvhi. Nbz zpv ibwf uif tusfohui gps uif ofyu dibncfs. Up gjoe uif fyju zpv gjstu xjmm offe up vuufs nbhjd xpset uifsf.”

The plain text came out to be this for key2:

“Be wary of the next chamber, there is very little joy there. Speak out the password "the_cave_man_be_pleased" to go through. May you have the strength for the next chamber. To find the exit you first will need to utter magic words there.”

Therefore key2 worked and we broke the cipher.