Q1 Commands

5 Points

List the commands was used in this level?

enter, put, back, enter, pluck, back, give, back, back, thrnxxtzy, read

Q2 Cryptosystem

10 Points

What cryptosystem was used in the game to reach the password?

The cryptosystem used is a Substitution-Permutation Cipher, which involves applying a permutation once and then using mono-alphabetic substitution to decode the ciphertext.

Q3 Analysis

30 Points

What tools and observations were used to figure out the cryptosystem and the password? (Explain in less than 1000 lines)

As there is another chamber there and we have to investigate, so "go/enter" is the command. Then there we saw a large hole and a small hole where something is shiny that small hole where we can barely put our hands. So we tried the "put" command to put our hands inside the hole but someone bites our hand. then the "back" command is given. Then to go to the underground chamber "enter" command was given. After that we finally found the smelly mushrooms, then upon several tries and investigation, we found that we have to pluck the mushrooms, hence, the "pluck" command was used. Then we remembered someone biting our hands, hence we got the idea that we have to give this mushroom to that thing. Hence, after plucking we came back to the chamber where the small hole was, using the "back" command. Then using the "give" command we gave the mushrooms. There we got the magic word "thrnxxtzy" from the spirit, to see the hidden door which lies in the main chamber. Hence, we used twice "back" command and there we spell this magic word i.e. "thrnxxtzy" command is used, then used the "read" command to read the text written in the glass panel.

First, we carried out the frequency analysis of each character, with the code used in Assignment 1. The frequency of characters resembles the standard frequencies of alphabets, hence substitution cipher might be used. However, when we tried searching for some frequent bigrams/trigrams, we were not able to find something non-trivial. This suggested that substitution cipher might not be employed solely. Hence we assumed that some Permutation might also be employed.
Analysis of the number of letters in the ciphertext, omitting the spaces and special characters, gave that the message had 284 characters.

Permutation and substitution are commutative actions. So we will try to figure out the permutation cipher before.
To begin with, we check for the permutation in blocks of 3. we took all the possible permutations of 3. Then, we permuted the whole string in blocks of 3 for all permutations and printed the highest frequency of bigrams in each permutation. The highest we were able to get is 2.8%.
We did the same for blocks of 4. The highest we got this time was 2.1%.
Next, we wanted to check in the blocks of 5. Since the word count is 284. We only took 280 letters and left the last 4 as it is. The highest frequency of any bigram we got was 5.4%. This is

much more promising than before. We got 5.4% for the permutation map {0, 1 ,2 ,3 ,4} -> {3, 2, 4, 0, 1}.
We also tried for blocks of 6. Again, the highest bigram frequency we got was 2.8%, which is much lower than what we go for 5.
Hence, considering the block permutation of 5 with key mentioned before and trying to figure out the substitution cipher.

We got the following text after deciphering the permutation (the last 4 alphabets were placed as it is along with the spaces and punctuations):

jnvqmvn ws afcl ewpv rctt jv jtvllvp jx afv lidvqmx lhcnca nvlcpcyg cy afv fwtv. gw qfvqp, qyp scyp q rqx ws jnvqmcyg afv lhvtt wy fcu eqla jx afv vbct kqssqn. afv lhcnca ws afv eqbv uqy cl qtrqxl rcaf xwd. scyp afv uqgce rqyp afqa rctt tva xwd wda ws afv eqbvl. ca rwdtp uqmv xwd q uqgcecqy, yw tvll afqy kqssqn! lhvqm afv hqllrwnp afv_uqgce_ws_rqyp aw gw afnwdgf.

First, we observed that the letter ‘q’ occurs frequently and it even occurs as a single-letter word. It can be considered to be the letter ‘a’. Then another very frequent trigram was found was ‘afv’. 'af' is also the most frequent bigram. The highest frequency bigram is 'th' in English Alphabet. Hence, 'afv' is considered to be ‘the’. So we replaced a,f, and v with t,h and e respectively.
Next, we assumed that the word before ‘afv_uqgce_ws_rqyp’ can be ‘password’ as it also contains ‘ll’ which can be substituted with ‘ss’. So we substituted h,l,r,w,n,p with p,s,w,o,r,d, respectively.
Further, we observed another frequent bigram where ‘o’ was followed by ‘s’. So we substituted ‘s’ with ‘f’, considering the bigram to be ‘of’. Next we found the word ‘eode’ and assumed this to be ‘code’, hence replacing ‘e’ with ‘c’. Then, we made ‘thcs’ to ‘this’, replacing ‘c’ with ‘i’. In ‘witt’, ‘tt’ can be assumed to be ‘ll’, so ‘t’ is substituted with ‘l’. Then in ‘thay’, ‘y’ was replaced with ‘n’.
So analysing more words and following the same pattern we replaced j, m, x, u, b, k, d, and I with b, k, y, m, v, j, u and q respectively, while ‘g’ remains unchanged. The word 'jaffar' is a bit

ambiguous an it can also be 'zaffar'. Hence, the final alphabet map is
'abcdefghijklmnopqrstuvwxyz' -> 'tviuchgpqbjskr?dawflmeoyn?'
Hence we obtained the plaintext:
breaker of this code will be blessed by the squeaky spirit residing in the hole. go ahead, and find a way of breaking the spell on him cast by the evil jaffar. the spirit of the caveman is always with you. find the magic wand that will let you out of the caves. it would make you a magician, no less than jaffar! speak the password the_magic_of_wand to go through.

The final password is : the_magic_of_wand

Q4 Password

5 Points

What was the final command used to clear this level?

the_magic_of_wand

Q5 Codes

0 Points

Upload any code that you have used to solve this level.

Assignment_3.ipynb

Download

In [318]:

```
# ciphertext
ciphertext = """qmnjvsa nv wewc flct vprj tj tvvplvl fv xja vqildhc
```

```
xmlnvc nacyclpa fc gyt vfvw. fv wgqyp, pqq pqcs y wsq
rx qmnjvafy cgv tlvhf cw tyl aeuq fv xja tkbv cqnsqs.
lhf avawnc cv eas fuqb qvq tc yllrqr xxwa cfy. psdc uqf
avrqc gefq pyat trac xwv taa wwd dv eas flcbq. vd trawm
vupq quw x decgqcwt, yq yafl vlqs yqklhq! snafq vml
lhvqpawr nqg_vfusr_ec_wawy qp fn wgawdgf."""
```

In [319]:

```
#count the number of alphabets in ciphertext
count=0
for i in ciphertext:
    if(i.isalpha()):
        count+=1
print(count)
```

```
284
```

In [320]:

```
#cipher is ciphertext with only alphabets
cipher=""
for i in ciphertext:       # i holds each character in String s for
every iteration of loop
    if(i.isalpha()):
        cipher=cipher+i
print(cipher)
```

```
qmnjvsanvwewcflctvprjtjtvvplvlfvxjavqildhcxmlnvcnacyclpafcgytvfvwfvwgqyp
```

In [321]:

```
#print frequency of each letter
import operator

temp={}
```

```
for char in ciphertext:
    if char.isalpha():
        temp[char.lower()]=temp.get(char.lower(),0)+1
res=dict(sorted(temp.items(),
key=operator.itemgetter(1),reverse=True))
for key,value in res.items():
    print("{} -> {:.1f}%".format(key, value/len(ciphertext)*100))
```

```
q -> 8.2%
v -> 7.9%
a -> 6.3%
c -> 6.0%
w -> 5.2%
f -> 5.2%
l -> 4.7%
t -> 3.6%
y -> 3.6%
s -> 3.0%
p -> 3.0%
n -> 2.7%
r -> 2.5%
x -> 2.2%
g -> 2.2%
e -> 1.9%
d -> 1.9%
j -> 1.6%
u -> 1.6%
m -> 1.4%
h -> 1.4%
b -> 0.8%
k -> 0.5%
i -> 0.3%
```

```
In [322]: import numpy as np
```

```
from itertools import permutations
from collections import Counter
```

In [323]:

```
#takes string and block permutation and returns the permuted string
def blockpermute(str, p):
    l=len(p)
    res=""
    i=0
    for ik in range(len(str)):
        temp=str[i:i+l]
        for j in range(l):
            res=res+temp[p[j]]
        i=i+l
        if i>=len(str):
            break

    return res
```

In [344]:

```
perms=np.array(list(permutations('01234'))).astype(int)
c=cipher[0:280]       #taking only the multiple of 5 for checking
for p in perms:
    res=blockpermute(c,p)

    cnt = Counter(res[x:x+2] for x in range(len(c) - 1))

    res=dict(sorted(cnt.items(),
key=operator.itemgetter(1),reverse=True))

    for key,value in res.items():
        if value>10:
            print("{} {} -> {:.1f}%".format(p[0:5],key,
value/len(c)*100))
```

```
            break
        cnt.clear()
```

```
[1 0 4 2 3] fa -> 4.6%
[2 4 0 1 3] af -> 3.9%
[3 1 0 4 2] fa -> 3.9%
[3 2 4 0 1] af -> 5.4%
```

In [327]:

```
permuted_cipher = blockpermute(c,[3, 2, 4, 0, 1])
print(len(permuted_cipher))
```

```
280
```

In [345]:

```
#sub is permuted cipher along with the spaces and punctuations like
original cipher
i=0
sub=""
for s in ciphertext:
    if s.isalpha() and i<280:
        sub=sub+permuted_cipher[i]
        i+=1
    else:
        sub+=s
print(sub)
```

```
jnvqmvn ws afcl ewpv rctt jv jtvllvp jx afv lidvqmx
lhcnca nvlcpcyg cy afv fwtv. gw qfvqp, qyp scyp q rqx
ws jnvqmcyg afv lhvtt wy fcu eqla jx afv vbct kqssqn.
afv lhcnca ws afv eqbv uqy cl qtrqxl rcaf xwd. scyp afv
uqgce rqyp afqa rctt tva xwd wda ws afv eqbvl. ca rwdtp
uqmv xwd q uqgcecqy, yw tvll afqy kqssqn! lhvqm afv
```

```
hqllrwnp afv_uqgce_ws_rqyp aw gw afnwdgf.
```

In [346]:

```
#substitution same as assignment 1
alphabet = 'abcdefghijklmnopqrstuvwxyz'
key = 'tviuchgpqbjskr?dawflmeoyn?'
def decrypt(ciphertext, key, alphabet):
    keyMap = dict(zip(alphabet, key))
    return ''.join(keyMap.get(c, c) for c in ciphertext)

plaintext = decrypt(sub, key, alphabet)

print(plaintext)
```

```
breaker of this code will be blessed by the squeaky
spirit residing in the hole. go ahead, and find a way
of breaking the spell on him cast by the evil jaffar.
the spirit of the cave man is always with you. find the
magic wand that will let you out of the caves. it would
make you a magician, no less than jaffar! speak the
password the_magic_of_wand to go through.
```

Q6 Group name

0 Points

ela