Q1 Commands
5 Points

List the commands used in the game to reach the first ciphertext.

go, read, enter, read

Q2 Cryptosystem
5 Points

What cryptosystem was used in this level?

Substitution Cipher (which is a symmetric cryptosystem) along with text rotation by 10 places to the left.

Q3 Analysis
25 Points

What tools and observations were used to figure out the cryptosystem? (Explain in less than 100 words)

Tools used: Frequency analysis using Expected Letter frequency distribution table for English text and Expected Bigram frequency distribution table for English text (https://bit.ly/3cIfg7I), python code.

Observations:
1. Occurrence of 3 single-letter words viz. 'P', 'Y' and 'A' raises doubts on placement of whitespaces since in regular vocabulary the only single-letter words used are 'I' and 'A'. From this observation, it is likely that the cipher text has been rotated.

2. The most frequently occurring letter in the ciphertext is 'Y', whereas according to expected frequency analysis, the most frequently occurring letter in English language is 'E', hence 'Y' is

very likely to be 'E'.

3. The most frequently occurring bigram in the ciphertext, 'ME' is very likely to be the most frequently occurring bigram in English language 'TH' (from Bigram Frequency Distribution Table of English language).

4. The second most frequently occurring bigram in ciphertext is 'EY', which is highly likely to be the second most frequently occurring bigram in English language which is 'HE'. Similarly, 'YS', third most frequently occurring bigram in cipher text will translate to 'E _ ' because we know that 'Y'='E'. Thus, we assume 'YS' = 'ER', because 'ER' is the expected most frequently occurring bigram starting with 'E' in any English text.

5. For 'YA', we need to find only 'A' because we have already assumed 'Y'='E'. From the bigram frequency distribution table, the next most frequently occurring combination starting with 'E' is 'ES'. Hence, 'A' is likely to be 'S'. The next most frequently occurring bigram in cipher text is 'WA', where 'A'='S'. Again, from the bigram frequency distribution table we can observe that the next most frequently occurring bigram in English with second alphabet as 'S' is 'IS'. Therefore, 'W' is most likely to be 'I'.

6. The word "IHTEREST" is now visible which gives 'H' = 'N'.

7. From the phrase "THE REI SNGT HINRG TI NTEREST IN THE" we can guess that it must be "THERE IS NOTHING OF INTEREST IN THE". This gives 'G'='O', 'R'='G' and 'T'='F'.

8. From this point onward, other words also start unravelling themselves, Ex: the cipher text "JORE INTE RE STIN GTHPNTHISON E" is most likely to be "MORE INTERESTING THAN THIS ONE". This gives 'J'='M' and 'P'='A'.

9. Cipher text "IN THEIHAMOER" can be translated to "IN THE CHAMBER". Thus, 'I'='C' and 'O'='B'.

10. The phrase "A SXO NCANSE E" is most likely to be "AS YOU CAN SEE". This gives 'X'='Y' and 'N'='U'.

11. "S UB S TITUTIONCIFHER" must be "SUBSTITUTION CIPHER". Therefore, 'F'='P'.

12. The word "PASSVOR U" must be "PASSWORD" and "DUOTES" must be "QUOTES" which gives 'V'='W', 'U'='D' and 'D'='Q'.

13. The words "CABES" and "KATER" are most likely "CAVES" and "LATER" respectively, which gives 'B'='V' and 'K'='L'.

14. The phrase "CH AMB ERSVI LLBE" is " CHAMBERS WILL BE", thus 'V'='W'.

15. All the digits have been shifted by 4 places. Reason: There's a sentence in the decrypted text which says "digits have been shifted by 8 places", however this 8 present in the deciphered text is also shifted. Thus, we can say that all the digits in original text are shifted by 'x' places such that digit 'x' in original text is mapped to ciphertext '8'. Therefore, the equation is (x+x)= 8 mod 10, this gives two possible values of 'x' i.e. 4 and 9. After trying out both the possibilities, we get 4 as the correct value. Thus, the encoding used for digits is: (Plaintext + 4) mod 10 = Ciphertext.

16. Special characters (, . !) appear to be unchanged.

Q4 Mapping

10 Points

What is the plaintext space and ciphertext space? What is the mapping between the elements of plaintext space and the elements of ciphertext space? (Explain in less than 100 words)

After substituting the letters and digits using the encoding stated in observations, the obtained text makes sense and we get fully formed words after rotating the cipher text by 10 places to the right, while keeping spaces fixed (i.e. whitespaces are not considered in rotation).

Ciphertext Space, C = "wsam ie pjo ysgtm eyipbya .P axg niphay

y,mey syw ahgm ewhrg tw hmysyam wh meyiepjoys .Ag jygtmeyk pmys ie pjo ysavw kkoyjgsywhmy sy amwh rmephmewagh y!Me yigu ynay utgsmew ajya apr ywap awjfkya no a mwmnmwghiwfeyswhve wieuwr wm aepby oyyhae wtmyuox8 fkpiya. Me y fpaavgs uwa mxSrN03u wddvwmegnmmey dngmya.Mew awameyt"

Plaintext Space (after substitution and rotation), M = "This is the first chamber of the caves. As you can see, there is nothing of interest in the chamber. Some of the later chambers will be more interesting than this one! The code used for this message is a simple substitution cipher in which digits have been shifted by 4 places. The password is tyRgU69diqq without the quotes."

Mapping from Ciphertext to Plaintext:
Y-->E, M-->T, E-->H, S-->R, A-->S, W-->I, H-->N, G-->O, R-->G, T-->F, P-->A, J-->M, I-->C, O-->B, X-->Y, N-->U, F-->P, V-->W, U-->D, D-->Q, B-->V, K-->L, 8-->4, 0-->6, 3-->9

Here Y->E was obtained using frequency analysis with the expected letter frequency distribution table. ME-->TH, EY-->HE, YS-->ER, YA-->ES, WA-->IS were obtained using frequency analysis with the expected bigram frequency distribution table and other mappings were obtained using observations as explained in Q3.

Q5 Password

5 Points

What was the final command used to clear this level?

tyRgU69diqq

Q6 Codes

0 Points

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

▾ Decrypt.ipynb Download

In [13]:
```
old_str = "wsam ie pjo ysgtm eyipbya .P axg
niphay y,mey syw ahgm ewhrg tw hmysyam wh
meyiepjoys .Ag jygtmeyk pmys ie pjo ysavw
kkoyjgsywhmy sy amwh rmephmewagh y!Me yigu
ynay utgsmew ajya apr ywap awjfkya no a
mwmnmwghiwfeyswhve wieuwr wm aepby oyyhae
wtmyuox8 fkpiya. Me y fpaavgs uwa mxSrN03u
wddvwmegnmmey dngmya.Mew awameyt"
string=''
for ch in old_str:
    if(ch==' '):
        continue
    elif(ch.isalpha()):
        string+=ch.upper()
    else:
        string+=ch
print("Cipher text:")
print(string)
```

```
Cipher text:
WSAMIEPJOYSGTMEYIPBYA.PAXGNIPHAYY,MEYSYWAHGMEWHRGT
```

In [14]:
```
dic={}
for char in string:
    if char.isalpha():
        if dic.get(char)==None:
            dic[char]=1
        else:
            dic[char]+=1
print("Frequency distribution in cipher
text")
for k,v in sorted(dic.items(), key = lambda
x: x[1], reverse = True):
    print(k,v/len(string)*100)
```

```
Frequency distribution in cipher text
Y 13.48314606741573
M 10.486891385767791
A 10.112359550561797
W 9.363295880149813
E 8.239700374531834
G 5.2434456928838955
S 4.868913857677903
P 4.868913857677903
H 4.49438202247191
I 3.3707865168539324
J 2.6217228464419478
O 2.6217228464419478
N 2.6217228464419478
T 2.247191011235955
U 2.247191011235955
R 1.8726591760299627
K 1.8726591760299627
V 1.4981273408239701
F 1.4981273408239701
X 1.1235955056179776

D 1.1235955056179776
B 0.7490636704119851
```

In [15]:

```
bi_dict={}
i=0
count=0
for i in range(0,len(string)-1):
    if(string[i].isalpha() and
string[i+1].isalpha()):
        bigram=string[i] +  string[i+1]

bi_dict[bigram]=bi_dict.get(bigram,0)+1
        count+=1
print("Bigram frequency distribution in
cipher text")
for k,v in sorted(bi_dict.items(), key =
lambda x: x[1], reverse = True):
    print(k,v,(v/count)*100)
```

```
Bigram frequency distribution in cipher text
ME 14 5.622489959839357
EY 9 3.614457831325301
YS 8 3.2128514056224895
YA 7 2.8112449799196786
WA 7 2.8112449799196786
AM 6 2.4096385542168677
EW 6 2.4096385542168677
WH 6 2.4096385542168677
EP 5 2.0080321285140563
OY 5 2.0080321285140563
MY 5 2.0080321285140563
IE 4 1.6064257028112447
SY 4 1.6064257028112447
HM 4 1.6064257028112447
MW 4 1.6064257028112447
PJ 3 1.2048192771084338
JO 3 1.2048192771084338
GT 3 1.2048192771084338
TM 3 1.2048192771084338
YI 3 1.2048192771084338
PA 3 1.2048192771084338
YW 3 1.2048192771084338
GS 3 1.2048192771084338
WM 3 1.2048192771084338
UW 3 1.2048192771084338
SA 2 0.8032128514056224
IP 2 0.8032128514056224
PB 2 0.8032128514056224
BY 2 0.8032128514056224
GN 2 0.8032128514056224
PH 2 0.8032128514056224
HA 2 0.8032128514056224
AY 2 0.8032128514056224
YY 2 0.8032128514056224
GM 2 0.8032128514056224
HR 2 0.8032128514056224
AG 2 0.8032128514056224
JY 2 0.8032128514056224
KP 2 0.8032128514056224
AV 2 0.8032128514056224
VW 2 0.8032128514056224
GH 2 0.8032128514056224
YU 2 0.8032128514056224
AA 2 0.8032128514056224
```

```
AP 2 0.8032128514056224
AW 2 0.8032128514056224
FK 2 0.8032128514056224
NM 2 0.8032128514056224
AE 2 0.8032128514056224
WS 1 0.4016064257028112
MI 1 0.4016064257028112
SG 1 0.4016064257028112
AX 1 0.4016064257028112
XG 1 0.4016064257028112
NI 1 0.4016064257028112
AH 1 0.4016064257028112
HG 1 0.4016064257028112
RG 1 0.4016064257028112
TW 1 0.4016064257028112
GJ 1 0.4016064257028112
YG 1 0.4016064257028112
YK 1 0.4016064257028112
PM 1 0.4016064257028112
SI 1 0.4016064257028112
WK 1 0.4016064257028112
KK 1 0.4016064257028112
KO 1 0.4016064257028112
YJ 1 0.4016064257028112
JG 1 0.4016064257028112
RM 1 0.4016064257028112
HY 1 0.4016064257028112
IG 1 0.4016064257028112
GU 1 0.4016064257028112
UY 1 0.4016064257028112
YN 1 0.4016064257028112
NA 1 0.4016064257028112
UT 1 0.4016064257028112
TG 1 0.4016064257028112
SM 1 0.4016064257028112
AJ 1 0.4016064257028112
PR 1 0.4016064257028112
RY 1 0.4016064257028112
WJ 1 0.4016064257028112
JF 1 0.4016064257028112
KY 1 0.4016064257028112
AN 1 0.4016064257028112
NO 1 0.4016064257028112
OA 1 0.4016064257028112
MN 1 0.4016064257028112
WG 1 0.4016064257028112
HI 1 0.4016064257028112
IW 1 0.4016064257028112
WF 1 0.4016064257028112
FE 1 0.4016064257028112
SW 1 0.4016064257028112
HV 1 0.4016064257028112
VE 1 0.4016064257028112
WI 1 0.4016064257028112
EU 1 0.4016064257028112
WR 1 0.4016064257028112
RW 1 0.4016064257028112
MA 1 0.4016064257028112
YO 1 0.4016064257028112
YH 1 0.4016064257028112
WT 1 0.4016064257028112
UO 1 0.4016064257028112
OX 1 0.4016064257028112
PI 1 0.4016064257028112
IY 1 0.4016064257028112
```

```
YF 1 0.4016064257028112
FP 1 0.4016064257028112
VG 1 0.4016064257028112
SU 1 0.4016064257028112
MX 1 0.4016064257028112
XS 1 0.4016064257028112
SR 1 0.4016064257028112
RN 1 0.4016064257028112
WD 1 0.4016064257028112
DD 1 0.4016064257028112
DV 1 0.4016064257028112
EG 1 0.4016064257028112
MM 1 0.4016064257028112
YD 1 0.4016064257028112
DN 1 0.4016064257028112
NG 1 0.4016064257028112
YT 1 0.4016064257028112
```

In [16]:

```
deciphered = ""
for ch in string:
    if ch=='D':
        ch='Q'
    elif ch=='U':
        ch='D'
    elif ch=='N':
        ch='U'
    elif ch=='H':
        ch='N'
    elif ch=='E':
        ch='H'
    elif ch=='V':
        ch='W'
    elif ch=='B':
        ch='V'
    elif ch=='O':
        ch='B'
    elif ch=='I':
        ch='C'
    elif ch=='F':
        ch='P'
    elif ch=='Y':
        ch='E'
    elif ch=='X':
        ch='Y'
    elif ch=='T':
        ch='F'
    elif ch=='M':
        ch='T'
    elif ch=='W':
        ch='I'
    elif ch=='G':
        ch='O'
    elif ch=='R':
        ch='G'
    elif ch=='S':
        ch='R'
    elif ch=='A':
        ch='S'
    elif ch=='P':
        ch='A'
    elif ch=='J':
        ch='M'
    elif ch=='K':
```

```
            ch='L'
        elif ch.isdigit():
            ch=str((int(ch)+6)%10)
        deciphered+=ch
    print("Deciphered text:")
    print(deciphered)
```

```
Deciphered text:
IRSTCHAMBEROFTHECAVES.ASYOUCANSEE,THEREISNOTHINGOF
```

In [ ]:

Assignment 1

● GRADED

GROUP

SHRUTI SHARMA
DEEKSHA ARORA
SAMBHRANT MAURYA
✎ View or edit group

TOTAL POINTS

45 / 50 pts

QUESTION 1

Commands — **5** / 5 pts

QUESTION 2

Cryptosystem — **5** / 5 pts

QUESTION 3

Analysis — **25** / 25 pts

QUESTION 4

Mapping — R **5** / 10 pts

QUESTION 5

Password **5** / 5 pts

QUESTION 6

Codes **0** / 0 pts