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 (mono-alphabetic in nature)

Q3 Analysis

25 Points

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

Tools:
(i) Used python script to check whether the given ciphertext is encrypted with "SHIFT CIPHER (CAESAR CIPHER)" or not.
(ii) Used python script (attached in answer 6) to find frequency of each letter and bigrams in the ciphertext.
(iii) Used table showing letter frequencies (unigram, bigram) in English language from the lecture slides and internet.

Observations:
1. We used python script to check if the ciphertext is encrypted with shift cipher or not. We found out that none of the 26 possibilities resulted in a meaningful text. Hence the possibility of encryption using shift cipher was rejected.

2. Then we proceed to check whether it is encrypted with Affine cipher and substitution cipher or not using Frequency analysis. The key in the mono-alphabetic substitution cipher defines a map from each letter of the plaintext alphabet to some (only one)letter of the ciphertext alphabet, where the map can be arbitrary subject only to the constraint that it be one-one so that decryption is possible. As a result, the key space contains all of the alphabet's bijections or permutations. When using English alphabets, the key Space is of size 26!=26*25*.....*1 or approximately 2^88, making brute-force attack impossible. Hence we go for frequency analysis which is going to utilize the statistical patterns of alphabets in English language.

3. If frequency analysis were not of work , we planned to hypothesize it as Vignere Cipher and decrypt it using Kasiski test and index of coincidence.

4. As we were doing Ciphertext only attack , these were all for the pre-planned as Encryption scheme like Hill Cipher , LFSR stream cipher etc are difficult to break with a ciphertext only attack.

Frequency Analysis-

5. The presence of three single-letter words, C H K, casts doubt on the placement of the spaces, because the only single-letter words used in regular English vocabulary are I and A. We deduced from this observation that there must be intentional spaces or a lack of spaces between words. Also, the last line does not end with a full stop, raising the possibility that the ciphertext has been rotated by ten places.

6. Because C appears significantly more frequently than any other ciphertext character in this ciphertext, we can hypothesize that d_k(C)=E. Looking at the bigram, we can see that the occurrence of FI is the highest, followed by the occurrence of IC. Because the most common bigrams in English are TH and HE, we hypothesize that d_k(FI)=TH and d_k(IC)=HE.

7. We will be seeing if it is encrypted with Affine Cipher. Mapping A to 0, B to 1, ...Z to 25, and from the above hypothesis , we got e_k(4)=2 and e_k(19)=5. As e_k(x)=ax+b, where a and b are unknown, two linear equation is as follows:

4a+b=2mod26 and 19a+b=5 mod26

This system has a unique solution as a=7 and b=0(in Z_26), which is a legal key as gcd(a,26)=1. Then, decrypting using x= 7-1x mod 26 i.e x=15y mod 26, we got no meaningful plaintext!. Hence we chose to check for Substitution Cipher.

8. At this point, substituting using above hypothesis , the partially decrypted ciphertext looks like :

OMKt Ph HDN eMGEt hePHSeK .H KRG VPHQKe e, the MeO KQGt hOQAG EO QteMeKt OQ thePhHDN eM .KG DeGEtheU HteM Ph HDN eMKLO UUNeDGMe OQte Me KtOQ AthHQthOKGQ e!th ePGY eVKe YEG MthO KDeK KHA eOKH KODJUeK VN K tOtVtO GQPOJheMOQLh OPhYOA Ot KhHSe NeeQKh OEte YNR2 JUHPeK. th e JHKKLGM YOK OMXR9V1X YA tLOthGVtthe XVGteK. thO KOKtheE

9. As of now, c is e, and the most common bigrams in the English language beginning with e is er and es. Assuming that d_k(CK) = ES and d_k(CM) = ER, the ciphertext now looks like:
Orst Ph HDN erGEt hePHSes .H sRG VPHQse e, the reO sQGt hOQAG EO Qterest OQ thePhHDN er .sG DeGEtheU Hter Ph HDN ersLO UUNeDGre OQte re stOQ AthHQthOsGQ e!th ePGY eVse YEG rthO sDes sHA eOsH sODJUes VN s tOtVtO GQPOJherOQLh OPhYOA Ot shHSe NeeQsh OEte YNR2 JUHPes. th e JHssLGr YOs OrXR9V1X YA tLOthGVtthe XVGtes. thO sOstheE

10. Taking a look at the phrase “se e, the reO s” which is more likely to be "see, there is," the possible substitution for O is I. the ciphertext now looks like
irst Ph HDN erGEt hePHSes .H sRG VPHQse e, the rei sQGt hiQAG Ei Qterest iQ thePhHDN er .sG DeGEtheU Hter Ph HDN ersLi UUNeDGre iQte re stiQ AthHQthisGQ e!th ePGY eVse YEG rthi sDes sHA eisH siDJUes VN s titVti GQPiJheriQLh iPhYiA it shHSe NeeQsh iEte YNR2 JUHPes. th e JHssLGr Yis irXR9V1X YA tLithGVtthe XVGtes. thi sistheE

11. By looking at “i Qterest “which is most likely the word "interest" leads us to the possible substitution for Q as N.

12. thi sistheEirst, probably "this is the first," which leads us to the possible substitution for E as F. The ciphertext now appears to be
irst Ph HDN erGft hePHSes .H sRGVPHnse e, the rei snGt hinAG fi nterest in thePhHDN er .sG DeGftheU Hter Ph HDN ersLi UUNeDGre inte re stin AthHnthisGn e!th ePGY eVse YfG rthi sDes sHA eisH siDJUes VN s titVti GnPiJherinLh iPhYiA it shHSe Neensh ifte YNR2 JUHPes. th e JHssLGr Yis irXR9V1X YA tLithGVtthe XVGtes. thi sisthef

13. The phrase “inte re stin AthHnthisGne” is possibly “interesting than this one”. Therefore the possible substitution for A is G, H Is A, G is O. The ciphertext after substituting now looks like:

irst Ph aDN eroft hePaSes .a sRo VPanse e, the rei snot hingo fi nterest in thePhaDN er .so DeoftheU ater Ph aDN ersLi UUNeDore inte re stin gthanthison e!th ePoY eVse Yfo rthi sDes sag eisa siDJUes VN s titVti onPiJherinLh iPhYig it shaSe Neensh ifte YNR2 JUaPes. th e JassLor Yis irXR9V1X Yg tLithoVtthe XVotes. thi sisthef

14. the phrase “so DeoftheU ater “ is more likely “some of the later”. Hence the possible substitution for D is M and U is L.
irst Ph amN eroft hePaSes .a sRo VPanse e, the rei snot hingo fi nterest in thePhamN er .so meofthel ater Ph amN ersLi llNemore inte re stin gthanthison e!th ePoY eVse Yfo rthi smes sag eisa simJles VN s titVti onPiJherinLh iPhYig it shaSe Neensh ifte YNR2 JlaPes. th e JassLor Yis irXR9V1X Yg tLithoVtthe XVotes. thi sisthef

15. “fo rthi smes sag eisa simJles VN s titVti onPiJher “ is more likely making the hypothesis of substituting J as P, and therefore P as C, and VN as UB.
irst ch amb eroft hecaSes .a sRo ucanse e, the rei snot hingo fi nterest in thechamb er .so meofthel ater ch amb ersLi llbemore inte re stin gthanthison e!th ecoY euse Yfo rthi smes sag eisa simples ub s tituti oncipherinLh ichYig it shaSe beensh ifte YbR2 places. th e passLor Yis irXR9U1X Yg tLithoutthe Xuotes. thi sisthef

16. The assignment task suggests that caSes should be caves. So d_k(S)=V,” sRo ucanse e” is “as you can see” leading d_k(R)=Y, “th ecoY euse Yfo rthi s “may be “the code used for this “ leading d_k(Y)=d, “l ater ch amb ersLi llbemore” , leads d_k(L)=W, and finally “Xuotes.” Is more likely “ quotes” and hence d_k(X)=Q. The final decryption leads us :
irst ch amb eroft hecaves .a syo ucanse e, the rei snot hingo fi nterest in thechamb er .so meofthel ater ch amb erswi llbemore inte re stin gthanthison e!th ecod euse dfo rthi smes sag eisa simples ub s tituti oncipherinwh ichdig it shave beensh ifte dby2 places. th e passwor dis irqy9u1q dg twithoutthe quotes. thi sisthef

17. Special characters appear to be unchanged!

18. We obtained the above plaintext after decrypting it with frequency analysis, which claims that the digits are shifted by "2" places. However, because 2 is a digit, it is obvious that 2 is also encrypted by some shifting. _Assume the number that was shifted to 2 is X. Because X is the key here, we can assert that X is shifted by X places, resulting in 2. The problem is written as follows in mathematical notation: X+X=2mod10 (mod 10 because there are 10 digits only, aka 0,1,2,3,4,5,6,7,8,9).
The digits satisfying the above equation is 1 and 6.
Without loss of generality, let us assume that X=1. Then the method of decryption tends to finding two numbers Y and Z, such that Y+1=9mod10 and Z+1=1mod10. Therefore leading us Y=8 and Z=0.
For this case the decrypted password is: iRqy8U0qdgt
On submission, it showed incorrect. So we tried the other value of X.

19. For the 2nd case assume that X=6. Then the method of decryption tends to finding two numbers Y and Z, such that Y+6=9mod10 and Z+6=1mod10. Therefore leading us as Y=3 and Z=5. For this case the decrypted password is: iRqy3U5qdgt
On submission, it showed correct.

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)

Plaintext space and ciphertext space are the sets of strings composed of uppercase English alphabets, lowercase English alphabets , digits, spaces and punctuations.

CIPHERTEXT SPACE:
“omkf pi hdn cmgef icphsck .H krg vphqkc c,fic mco kqgf ioqag eo qfcmckf oq ficpihdn cm .Kg dcgeficu hfcm pi hdn cmklo uuncdgmc oqfc mc kfoq afihqfiokgq c!Fi cpgy cvkc yeg mfio kdck kha cokh kodjuck vn k fofvfo gqpojicmoqli opiyoa of kihsc nccqki oefc ynr2 juhpck. Fi c jhkklgm yok oMxr9V1x ya flofigvffic xvgfck. Fio kokfice”

PLAINTEXT SPACE (after substitution and rotation) :
“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 2 places. The password is iRqy3U5qdgt without the quotes.”

The following mapping is extracted from the explanation in question 3:
A→g, C→e, D→m, E→f, F→t, G→o, H→a, I→ h, J→p, K→s, L→w, M→r, N→b, O→i, P→c, Q→n, R→y, S→v, U→l, V→u, X→q, Y→d, 2→6, 9→3, 1→5

Q5 Password

5 Points

What is the final command used to clear this level?

iRqy3U5qdgt

Q6 Codes

0 Points

Upload any code that you have used to solve this level

Codes.ipynb Download

Original Ciphertext

In [22]:

```
ciphertxt = "omkf pi hdn cmgef icphsck .H krg vphqkc c,fic mco kqgf
ioqag eo qfcmckf oq ficpihdn cm .Kg dcgeficu hfcm pi hdn cmklo
uuncdgmc oqfc mc kfoq afihqfiokgq c!Fi cpgy cvkc yeg mfio kdck kha
cokh kodjuck vn k fofvfo gqpojicmoqli opiyoa of kihsc nccqki oefc
ynr2 juhpck. Fi c jhkklgm yok oMxr9V1x ya flofigvffic xvgfck. Fio
kokfice"
ciphertxt = ciphertxt.upper().replace(" ", "")
print("Cipher text:")
print(ciphertxt)
```

```
Cipher text:
OMKFPIHDNCMGEFICPHSCK.HKRGVPHQKCC,FICMCOKQGFIOQAGEOQFCMCKFOQFICPIHDNCM.KGDC
```

Shift Cipher: Try to check if decryption using shift cipher is meaningful

In [23]:

```
## SHIFT CIPHER ##
import string
alphabets = string.ascii_uppercase
for i in range(25):
    temp = ""
    for j in ciphertxt:
        ch = j
        if(ch.isalpha()):
            loc = (alphabets.index(ch) + i)%26
            temp+=alphabets[loc]
        else:
```

```
            temp+=ch
    print(temp)
```

```
OMKFPIHDNCMGEFICPHSCK.HKRGVPHQKCC,FICMCOKQGFIOQAGEOQFCMCKFOQFICPIHDNCM.KGDC
PNLGQJIEODNHFGJDQITDL.ILSHWQIRLDD,GJDNDPLRHGJPRBHFPRGDNDLGPRGJDQJIEODN.LHED
QOMHRKJFPEOIGHKERJUEM.JMTIXRJSMEE,HKEOEQMSIHKQSCIGQSHEOEMHQSHKERKJFPEO.MIFE
RPNISLKGQFPJHILFSKVFN.KNUJYSKTNFF,ILFPFRNTJILRTDJHRTIFPFNIRTILFSLKGQFP.NJGF
SQOJTMLHRGQKIJMGTLWGO.LOVKZTLUOGG,JMGQGSOUKJMSUEKISUJGQGOJSUJMGTMLHRGQ.OKHC
TRPKUNMISHRLJKNHUMXHP.MPWLAUMVPHH,KNHRHTPVLKNTVFLJTVKHRHPKTVKNHUNMISHR.PLIH
USQLVONJTISMKLOIVNYIQ.NQXMBVNWQII,LOISIUQWMLOUWGMKUWLISIQLUWLOIVONJTIS.QMJJ
VTRMWPOKUJTNLMPJWOZJR.ORYNCWOXRJJ,MPJTJVRXNMPVXHNLVXMJTJRMVXMPJWPOKUJT.RNKJ
WUSNXQPLVKUOMNQKXPAKS.PSZODXPYSKK,NQKUKWSYONQWYIOMWYNKUKSNWYNQKXQPLVKU.SOLK
XVTOYRQMWLVPNORLYQBLT.QTAPEYQZTLL,ORLVLXTZPORXZJPNXZOLVLTOXZORLYRQMWLV.TPML
YWUPZSRNXMWQOPSMZRCMU.RUBQFZRAUMM,PSMWMYUAQPSYAKQOYAPMWMUPYAPSMZSRNXMW.UQNM
ZXVQATSOYNXRPQTNASDNV.SVCRGASBVNN,QTNXNZVBRQTZBLRPZBQNXNVQZBQTNATSOYNX.VRON
AYWRBUTPZOYSQRUOBTEOW.TWDSHBTCWOO,RUOYOAWCSRUACMSQACROYOWRACRUOBUTPZOY.WSPC
BZXSCVUQAPZTRSVPCUFPX.UXETICUDXPP,SVPZPBXDTSVBDNTRBDSPZPXSBDSVPCVUQAPZ.XTQF
CAYTDWVRBQAUSTWQDVGQY.VYFUJDVEYQQ,TWQAQCYEUTWCEOUSCETQAQYTCETWQDWVRBQA.YURC
DBZUEXWSCRBVTUXREWHRZ.WZGVKEWFZRR,UXRBRDZFVUXDFPVTDFURBRZUDFUXREXWSCRB.ZVSF
ECAVFYXTDSCWUVYSFXISA.XAHWLFXGASS,VYSCSEAGWVYEGQWUEGVSCSAVEGVYSFYXTDSC.AWTS
FDBWGZYUETDXVWZTGYJTB.YBIXMGYHBTT,WZTDTFBHXWZFHRXVFHWTDTBWFHWZTGZYUETD.BXUT
GECXHAZVFUEYWXAUHZKUC.ZCJYNHZICUU,XAUEUGCIYXAGISYWGIXUEUCXGIXAUHAZVFUE.CYVU
HFDYIBAWGVFZXYBVIALVD.ADKZOIAJDVV,YBVFVHDJZYBHJTZXHJYVFVDYHJYBVIBAWGVF.DZWV
IGEZJCBXHWGAYZCWJBMWE.BELAPJBKEWW,ZCWGWIEKAZCIKUAYIKZWGWEZIKZCWJCBXHWG.EAXW
JHFAKDCYIXHBZADXKCNXF.CFMBQKCLFXX,ADXHXJFLBADJLVBZJLAXHXFAJLADXKDCYIXH.FBYX
KIGBLEDZJYICABEYLDOYG.DGNCRLDMGYY,BEYIYKGMCBEKMWCAKMBYIYGBKMBEYLEDZJYI.GCZY
LJHCMFEAKZJDBCFZMEPZH.EHODSMENHZZ,CFZJZLHNDCFLNXDBLNCZJZHCLNCFZMFEAKZJ.HDAZ
MKIDNGFBLAKECDGANFQAI.FIPETNFOIAA,DGAKAMIOEDGMOYECMODAKAIDMODGANGFBLAK.IEBA
```

Affine Cipher: Try to check if decryption using affine cipher is meaningful

In [24]:

```
### AFFINE Cipher #####
temp2 = ""
for j in ciphertxt:
    ch = j
    if(ch.isalpha()):
        loc = alphabets.index(ch)
        temp2+=alphabets[((loc*15)%26)]
    else:
        temp2+=ch
print(temp2)
```

```
CYUXRQBTNEYMIXQERBKEU.BUVMDRBGUEE,XQEYECUGMXQCGAMICGXEYEUXCGXQERQBTNEY.UMTE
```

Frequency Analysis: Unigrams

In [25]:

```
#### UNIGRAMS #########
freq = {i:ciphertxt.count(i)/len(ciphertxt)*100 for i in alphabets}
sorted(freq.items(), key = lambda x: x[1], reverse = True)
```

Out [25]:

```
[('C', 13.48314606741573),
 ('F', 10.486891385767791),
 ('K', 10.112359550561797),
 ('O', 9.363295880149813),
 ('I', 8.239700374531834),
 ('G', 5.2434456928838955),
 ('H', 4.868913857677903),
 ('M', 4.868913857677903),
 ('Q', 4.49438202247191),
 ('P', 3.3707865168539324),
 ('D', 2.6217228464419478),
 ('N', 2.6217228464419478),
 ('V', 2.6217228464419478),
 ('E', 2.247191011235955),
 ('Y', 2.247191011235955),
 ('A', 1.8726591760299627),
 ('U', 1.8726591760299627),
 ('J', 1.4981273408239701),
 ('L', 1.4981273408239701),
 ('R', 1.1235955056179776),
 ('X', 1.1235955056179776),
 ('S', 0.7490636704119851),
 ('B', 0.0),
 ('T', 0.0),
 ('W', 0.0),
 ('Z', 0.0)]
```

Frequency Analysis: Bigrams

In [26]:

```
##### BI-GRAMS #######
fr = {i+j:ciphertxt.count(i+j) for i in alphabets for j in alphabets
if ciphertxt.count(i+j)>0}
p =sum(fr.values())
for i in fr:
    fr[i] = float(fr[i]/p)*100
sorted(fr.items(), key = lambda x: x[1], reverse = True)
```

Out [26]:

```
[('FI', 5.64516129032258),
 ('IC', 3.6290322580645165),
 ('CM', 3.225806451612903),
 ('CK', 2.82258064516129),
 ('OK', 2.82258064516129),
 ('IO', 2.4193548387096775),
 ('OQ', 2.4193548387096775),
 ('FC', 2.0161290322580645),
 ('IH', 2.0161290322580645),
 ('KF', 2.0161290322580645),
 ('NC', 2.0161290322580645),
 ('FO', 1.6129032258064515),
 ('MC', 1.6129032258064515),
 ('PI', 1.6129032258064515),
 ('QF', 1.6129032258064515),
 ('CO', 1.2096774193548387),
 ('CP', 1.2096774193548387),
 ('DN', 1.2096774193548387),
 ('EF', 1.2096774193548387),
 ('GE', 1.2096774193548387),
 ('GM', 1.2096774193548387),
 ('HD', 1.2096774193548387),
 ('HK', 1.2096774193548387),
 ('KO', 1.2096774193548387),
 ('OF', 1.2096774193548387),
 ('AF', 0.8064516129032258),
 ('CC', 0.8064516129032258),
 ('CY', 0.8064516129032258),
 ('DC', 0.8064516129032258),
 ('GF', 0.8064516129032258),
 ('GQ', 0.8064516129032258),
 ('GV', 0.8064516129032258),
 ('HQ', 0.8064516129032258),
 ('HS', 0.8064516129032258),
 ('JU', 0.8064516129032258),
 ('KC', 0.8064516129032258),
 ('KG', 0.8064516129032258),
 ('KH', 0.8064516129032258),
```

```
('KI', 0.8064516129032258),
('KK', 0.8064516129032258),
('KL', 0.8064516129032258),
('LO', 0.8064516129032258),
('MK', 0.8064516129032258),
('OM', 0.8064516129032258),
('PH', 0.8064516129032258),
('QA', 0.8064516129032258),
('QK', 0.8064516129032258),
('SC', 0.8064516129032258),
('UH', 0.8064516129032258),
('VF', 0.8064516129032258),
('YO', 0.8064516129032258),
('AC', 0.4032258064516129),
('AG', 0.4032258064516129),
('AO', 0.4032258064516129),
('CD', 0.4032258064516129),
('CE', 0.4032258064516129),
('CG', 0.4032258064516129),
('CJ', 0.4032258064516129),
('CN', 0.4032258064516129),
('CQ', 0.4032258064516129),
('CU', 0.4032258064516129),
('CV', 0.4032258064516129),
('CX', 0.4032258064516129),
('DG', 0.4032258064516129),
('DJ', 0.4032258064516129),
('EG', 0.4032258064516129),
('EO', 0.4032258064516129),
('FF', 0.4032258064516129),
('FK', 0.4032258064516129),
('FL', 0.4032258064516129),
('FP', 0.4032258064516129),
('FV', 0.4032258064516129),
('GD', 0.4032258064516129),
('GY', 0.4032258064516129),
('HA', 0.4032258064516129),
('HF', 0.4032258064516129),
('HP', 0.4032258064516129),
('IG', 0.4032258064516129),
('IY', 0.4032258064516129),
('JH', 0.4032258064516129),
('JI', 0.4032258064516129),
('KD', 0.4032258064516129),
('KQ', 0.4032258064516129),
('KR', 0.4032258064516129),
('KV', 0.4032258064516129),
```

```
('LG', 0.4032258064516129),
('LI', 0.4032258064516129),
('MF', 0.4032258064516129),
('MG', 0.4032258064516129),
('MO', 0.4032258064516129),
('MP', 0.4032258064516129),
('MX', 0.4032258064516129),
('MY', 0.4032258064516129),
('NK', 0.4032258064516129),
('NR', 0.4032258064516129),
('OA', 0.4032258064516129),
('OD', 0.4032258064516129),
('OE', 0.4032258064516129),
('OG', 0.4032258064516129),
('OJ', 0.4032258064516129),
('OP', 0.4032258064516129),
('OU', 0.4032258064516129),
('PC', 0.4032258064516129),
('PG', 0.4032258064516129),
('PO', 0.4032258064516129),
('QC', 0.4032258064516129),
('QG', 0.4032258064516129),
('QL', 0.4032258064516129),
('QP', 0.4032258064516129),
('RG', 0.4032258064516129),
('UC', 0.4032258064516129),
('UN', 0.4032258064516129),
('UU', 0.4032258064516129),
('VG', 0.4032258064516129),
('VK', 0.4032258064516129),
('VN', 0.4032258064516129),
('VP', 0.4032258064516129),
('XR', 0.4032258064516129),
('XV', 0.4032258064516129),
('XY', 0.4032258064516129),
('YA', 0.4032258064516129),
('YC', 0.4032258064516129),
('YE', 0.4032258064516129),
('YN', 0.4032258064516129)]
```

Substitution Cipher using frequency analysis

In [27]:

```
plaintxt = ""
```

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

```
print("Deciphered text:")
print(plaintxt)
```

```
Deciphered text:
IRSTCHAMBEROFTHECAVES.ASYOUCANSEE,THEREISNOTHINGOFINTERESTINTHECHAMBER.SOME
```

Q7 Team Name

0 Points

Enciphered

Assignment 1

● GRADED

GROUP

Gargi Sarkar
Utkarsh Srivastava
Anindya Ganguly

View or edit group

TOTAL POINTS

50 / 50 pts

QUESTION 1

Commands — **5** / 5 pts

QUESTION 2

Cryptosystem — **5** / 5 pts

QUESTION 3

Analysis — **25** / 25 pts

QUESTION 4

Mapping — **10** / 10 pts

QUESTION 5

Password — **5** / 5 pts

QUESTION 6

Codes — **0** / 0 pts

QUESTION 7

Team Name — **0** / 0 pts