Q1 Commands
10 Points

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

go
back
read

Q2 Cryptosystem
10 Points

What cryptosystem was used in this level?

We used "Vigenere Cipher" cryptosystem to decrypt at this level.

Q3 Analysis
20 Points

What tools and observations were used to figure out the cryptosystem?

NOTE: Failing to provide proper analysis would result in zero marks for this assignment.

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 2 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.
3. 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.
4. After analyzing unigrams and bigrams frequency we were further sure that substitution cipher is not the encryption algorithm used in current scenario. Because none of the two letter and three letter words are repeated , We thought that the ciphertext may be encrypted by vignere cipher.
5. In our puzzle there was a question to count lines in horizontal direction from bottom to up and after counting in horizontal direction by moving in upward direction we got number of lines as follows: [9,2,9,2,5,5,2,2,2,1].
6. The above was our numeric key which we converted into letter key by applying a function which maps the numbers in the number key to the corresponding alphabets . For example A->0,B->1,C->2,D->3,E->4,F->5,..... X->23,Y->24,Z->25 and so on. We used a function which performs this task as follows :
7. (a) This function converts a list of numbers to a string of letters : numeric_to_letter_key(key):

Initialize an empty string to store the letter key , and Iterate over each number in the input
key.
(b) Convert each number to its corresponding letter using the modulo operator and the chr
function
(c) The formula (ord('A') + int(number) % 26) calculates the ASCII code of the letter
corresponding to the number.
(d)The chr function converts the ASCII code to the actual letter
letter_key += chr(ord('A') + int(number) % 26)
(e) Return the final letter key
8. So the letter key we got after applying this function was "JCJCFFCCCB". We used this key to
decipher vignere cipher. By using the key "JCJCFFCCCB" and using the decryption algorithm
as given in the code , the ciphertext :
" Kg fcwd qh vin pnzy hjcocnt, cjjwg ku wnth nnyvng kxa cjjwg. Urfjm xwy yjg rbbufqwi
"vjg_djxn_ofs_dg_rmncbgi" yq iq uqtxwlm. Oca zxw qcaj vjg
tctnplyj hqs cjn pjcv ejbvdnt. Yt hkpe cjn gcnv, aqv okauy bknn ongm vt zvvgs vcpkh bqtft
cjntj"
was succesfully decrypted as plaintext given below:
"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."

Q4 Decryption Algorithm
15 Points

Briefly describe the decryption algorithm used. Also mention the plaintext you deciphered. (Use less than 350 words)

The ciphertext of the question is "Kg fcwd qh vin pnzy hjcocnt, cjjwg ku wnth nnyvng kxa cjjwg. Urfjm xwy yjg rbbufqwi "vjg_djxn_ofs_dg_rmncbgi" yq iq uqtxwlm. Oca

zxw qcaj vjg tctnplyj hqs cjn pjcv ejbvdnt. Yt hkpe cjn gcnv, aqv okauy bknn ongm vt zvvgs vcpkh bqtft cjntj."

The decryption algorithm we used to decipher the vignere cipher is as follows:
1. Initialize an empty string plaintext to store the decrypted text.
2. Initialize a variable key_index to keep track of the current letter in the key. Set it to 0.
3. Initialize a variable key_len to store the length of the key. Set it to the length of the key string.
4. Iterate over each character char in the input cipher text cipher_text.
    1. If the character is an alphabet,
        1. Calculate the shift value by subtracting the ASCII value of 'A' from the ASCII value of
            the uppercase form of the current letter in the key.
    2. If the character is in lowercase,
        1. Calculate the decrypted character using the formula (ord(char) - ord('a') - shift + 26)
            % 26 + ord('a').
        2. Add the decrypted character to plaintext.
    3. If the character is in uppercase,
        1. Calculate the decrypted character using the formula (ord(char) - ord('A') - shift + 26)
            % 26 + ord('A').
        2. Add the decrypted character to plaintext.
        3. Increment key_index by 1 and take its modulo with key_len to ensure it stays within
            the range of the key length.
    4. If the character is not an alphabet,
        1. add the character to plaintext as it is.
5. Return plaintext.
6. In the main code, pass the plaintext and key to the vigenere_decrypt function and store the
   result in plaintext. The key we got after solving the caeman puzzle is "JCJCFFCCCB".
7. Print plaintext.
8. The plaintext we deciphered is "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.

Q5 Password
10 Points

What was the final command used to clear this level?

the_cave_man_be_pleased

Q6 Codes
0 Points

Upload any code that you have used to solve this level

Modern_Cryptology1 (1).ipynb Download

Shift Cipher: Trying to check whether shift cipher is meaningful or not

In [10]:
```
def bruteforce_shift_cipher(ciphertext):
    for i in range(1, 26):
        plaintext = ""
        for char in ciphertext:
            if char.isalpha():
                char_code = ord(char)
                if char.isupper():
                    char_code -= i
                    if char_code < ord('A'):
                        char_code += 26
                elif char.islower():
                    char_code -= i
                    if char_code < ord('a'):
                        char_code += 26
                plaintext += chr(char_code)
```

```
            else:
                plaintext += char
        print(f"Key: {i},
Plaintext: {plaintext}")
```

In [11]:

```
ciphertext ="Kg fcwd qh vin pnzy
hjcocnt, cjjwg ku wnth nnyvng kxa
cjjwg. Urfjm xwy yjg rbbufqwi
vjg_djxn_ofs_dg_rmncbgi yq iq
uqtxwlm. Oca zxw qcaj vjg tctnplyj
hqs cjn pjcv ejbvdnt. Yt hkpe cjn
gcnv, aqv okauy bknn ongm vt zvvgs
vcpkh bqtft cjntj."
bruteforce_shift_cipher(ciphertext)
```

```
Key: 1, Plaintext: Jf ebvc pg uhm omyx
Key: 2, Plaintext: Ie daub of tgl nlxw
Key: 3, Plaintext: Hd czta ne sfk mkwv
Key: 4, Plaintext: Gc bysz md rej ljvu
Key: 5, Plaintext: Fb axry lc qdi kiut
Key: 6, Plaintext: Ea zwqx kb pch jhts
Key: 7, Plaintext: Dz yvpw ja obg igsr
Key: 8, Plaintext: Cy xuov iz naf hfrq
Key: 9, Plaintext: Bx wtnu hy mze geqp
Key: 10, Plaintext: Aw vsmt gx lyd fdpo
Key: 11, Plaintext: Zv urls fw kxc econ
Key: 12, Plaintext: Yu tqkr ev jwb dbnm
Key: 13, Plaintext: Xt spjq du iva caml
Key: 14, Plaintext: Ws roip ct huz bzlk
Key: 15, Plaintext: Vr qnho bs gty aykj
Key: 16, Plaintext: Uq pmgn ar fsx zxji
Key: 17, Plaintext: Tp olfm zq erw ywih
Key: 18, Plaintext: So nkel yp dqv xvhg
Key: 19, Plaintext: Rn mjdk xo cpu wugf
Key: 20, Plaintext: Qm licj wn bot vtfe
Key: 21, Plaintext: Pl khbi vm ans used
Key: 22, Plaintext: Ok jgah ul zmr trdc
Key: 23, Plaintext: Nj ifzg tk ylq sqcb
Key: 24, Plaintext: Mi heyf sj xkp rpba
Key: 25, Plaintext: Lh gdxe ri wjo qoaz
```

Frequency Analysis : Unigrams

In [12]:

```
def
frequency_analysis(ciphertext):

    freq_dict = {}
```

```
    for char in ciphertext:
        if char.isalpha():
            if char in freq_dict:
                freq_dict[char] +=
1
            else:
                freq_dict[char] =
1

    freq_dict =
dict(sorted(freq_dict.items(),
key=lambda item: item[1],
reverse=True))

    print("Frequency:", freq_dict)
```

In [13]:

```
ciphertext = "Kg fcwd qh vin pnzy
hjcocnt, cjjwg ku wnth nnyvng kxa
cjjwg. Urfjm xwy yjg rbbufqwi
vjg_djxn_ofs_dg_rmncbgi yq iq
uqtxwlm. Oca zxw qcaj vjg tctnplyj
hqs cjn pjcv ejbvdnt. Yt hkpe cjn
gcnv, aqv okauy bknn ongm vt zvvgs
vcpkh bqtft cjntj."
frequency_analysis(ciphertext)
```

```
Frequency: {'n': 18, 'j': 18, 'c': 15,
```

Letter Pair Analysis

In [14]:

```
def
letter_pair_analysis(ciphertext):

    freq_dict = {}
    for i in
range(len(ciphertext)-1):
        if ciphertext[i].isalpha()
and ciphertext[i+1].isalpha():
            letter_pair =
ciphertext[i] + ciphertext[i+1]
            if letter_pair in
freq_dict:

freq_dict[letter_pair] += 1
            else:

freq_dict[letter_pair] = 1
```

```
    freq_dict =
dict(sorted(freq_dict.items(),
key=lambda item: item[1],
reverse=True))

    print("Frequency:", freq_dict)
```

In [15]:

```
ciphertext = "Kg fcwd qh vin pnzy
hjcocnt, cjjwg ku wnth nnyvng kxa
cjjwg. Urfjm xwy yjg rbbufqwi
vjg_djxn_ofs_dg_rmncbgi yq iq
uqtxwlm. Oca zxw qcaj vjg tctnplyj
hqs cjn pjcv ejbvdnt. Yt hkpe cjn
gcnv, aqv okauy bknn ongm vt zvvgs
vcpkh bqtft cjntj.."
letter_pair_analysis(ciphertext)
```

```
Frequency: {'cj': 5, 'nt': 4, 'xw': 3,
```

In [ ]:

Modern_Crypto_assignment2.ipynb

Download

In [1]:

```
def numeric_to_letter_key(key):
    letter_key = ""
    for number in key:
        letter_key += chr(ord('A')
+ int(number) % 26)
    return letter_key

numeric_key =
[9,2,9,2,5,5,2,2,2,1]
letter_key =
numeric_to_letter_key(numeric_key)
print("Letter key:", letter_key)
```

```
Letter key: JCJCFFCCCB
```

In [2]:

```
def decrypt(cipher_text, key):
    key_len = len(key)
    key_index = 0
    output_text = ""
    for char in cipher_text:
```

```
        if char.isalpha():
            shift =
ord(key[key_index].upper()) -
ord('A')
            if char.islower():
                output_text +=
chr((ord(char) - ord('a') - shift
+ 26) % 26 + ord('a'))
            else:
                output_text +=
chr((ord(char) - ord('A') - shift
+ 26) % 26 + ord('A'))
            key_index = (key_index
+ 1) % key_len
        else:
            output_text += char


    return output_text

plaintext = "Kg fcwd qh vin pnzy
hjcocnt, cjjwg ku wnth nnyvng kxa
cjjwg. Urfjm xwy yjg rbbufqwi
vjg_djxn_ofs_dg_rmncbgi yq iq
uqtxwlm. Oca zxw qcaj vjg tctnplyj
hqs cjn pjcv ejbvdnt. Yt hkpe cjn
gcnv,aqv okauy bknn ongm vt zvvgs
vcpkh bqtft cjntj."
key = "JCJCFFCCCB"

output_text = decrypt(plaintext,
key)
print(output_text)
```

```
Be wary of the next chamber, there is v
```

In [ ]:

Q7 Team Name

0 Points

d2ce09fd5842b342d5b3b66d10b6daef

Assignment 2

Graded

Group

MADHAV MAHESHWARI
RAJ KUMAR
GUNJ MEHUL HUNDIWALA
View or edit group

Total Points

60 / 65 pts

Question 1

Commands — **10** / 10 pts

Question 2

Cryptosystem — **10** / 10 pts

Question 3

Analysis — **15** / 20 pts

Question 4

Decryption Algorithm — R **15** / 15 pts

Question 5

Password — **10** / 10 pts

Question 6

Codes — **0** / 0 pts

Question 7

Team Name — **0** / 0 pts