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

Q3 Analysis
25 Points

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

We wrote a C++ code to implement the following logic for de cryption. Code is uploaded in the Q6's answer.

We found following reasons to beleive that spaces in the cip hered text are random and are not according the English La nguage:
1. Letters after full stop were capital which suggests that Engl ish formatting around punctuation is followed. But spaces in the message are uneven, in some places there were spaces before full stop and in some places there were after it. There are no consistency.
2. After the excalmation mark ("!") there is no space.
3. There are 3 single letter words in the ciphered message - 'P', 'a', and 'y'. Whereas in English language there is only two single letter words - 'A' and 'I'

Above points prompts us to remove the spaces from the me

ssage.

After removing spaces we found many combination of letter s are repeating in the message like "iepjoys" is repeated 3 ti mes, "mey" is repeated 8 times, "mewa" is repeated 3 times, "wa" is repeated 7 times, "gt" is repeated thrice, "whmysya m" is repeated twice and many more. This suggests that enc ryption method is substitution.

So we performed the frequency analysis on the cipher text a nd the top 7 records of the distribution are as follows:
y - 13.9535%
m - 10.8527%
a - 10.4651%
w - 9.68992%
e - 8.52713%
g - 5.42636%
s - 5.03876%

Since percentage of 'y' is 13.9% and there is exist a gap of 3% from the next highest frequency we assumed that 'e' is s ubstituted with 'y' (e->y) and 't' with 'm' (t->m). After this substi tution "mey" which is repeated 8 times becomes a 3 letter w ord which starts with 't' and ends with 'e'. Since "the" is the m ost common word in English language, we substitute 'e' with 'h' (e->h).

After this substitution we break the word "the" from the ciph er by adding spaces. Since in English language word "the" fo llows a preposition (in,on,at,of,for,or). The string "gt" preceed es the word "the" in this partially decrypted message twice, by bruteforcing among the preposition we found the substit ution 'o'->'g' and 'f'->'t' we guessed "wa" to be as "is" as it ma de "mewa" to the word "this" which is repeated 3 times and i nduces the preposition "is" in 4 other places. So we finalize d: 'i'->'w' and 's'->'a'.

Same proposition theory applied to "i_ the" which deciphers to "in the" and hence maps 'n'->'h'.

In the Frequency distribution table of English text next letter s are 'a' and 'o'. So we tried both of them for the letters 'g' an

d 'p' which are next in the frequency distribution of cipher te xt. After brute force we found the mapping - 'o'->'g' and 'a'->' p'.

Some of the interesting guesses are:
1. By looking at 'of inte_est' ,we can say the word is 'interest'. 'r' -> 's'.
2. By looking at the string "the re is not hin_ of interest", the actual sentence seems like "there is nothin_ of interest". 'not hin_' looks like 'nothing', So, 'g' -> 'r'.

After this it was pretty much easy guessing up the words to decipher the whole text.

Solving the substitution on the digits

"Digits have been shifted by 8 places." - the clause in the de ciphered text by substituting the alphabets. Since 8 is a digit, which means digits are not exactly shifted by 8 places as eig ht in the clause is also shifted. Let us consider that x is the n umber of places by which digits are shifted. So the clause be fore ciphered is like - "Digits have been shifted by x places."

Hence (x+x)mod10=8. This leads to two solution of x - either x=4 or x=9. By this two possible password exists: "tyRgU69di qq" and "tyRgU14diqq". We tried both of them in the game an d found "tyRgU69diqq" as the password.

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)

Plain Text after deciphering

This is the first chamber of the caves. As you can see, there i s nothing of interest in the chamber. Some of the later cham bers will be more interesting than this one ! The code used f

or this message is a simple substitution cipher in which digit s have been shifted by 4 places. The password is tyRgU69di qq without the quotes.

Plain Text Space

Plain_text_set = {a, b, c, d, e, f, g, h, i, l, m, n, o, p, q, r, s, t, u, v, w, y, 8, 6, 9}

Cipher Text Space

Cipher_text_set = {a, b, d, e, f, g, h, i, j, k, m, n, o, p, r, s, t, u, v, w, x, y, 0, 8, 3}

Mapping

Plain text -> Cipher text

e -> y
t -> m
s -> a
i -> w
h -> e
o -> g
r -> s
a -> p
n -> h
c -> i
u -> n
m -> j
b -> o
d -> u
f -> t
l -> k
g -> r
p -> f
w -> v
q -> d
y -> x
v -> b
6 -> 0
9 -> 3

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.

assign1.cpp Download

```
// Color Codes
// [0;31m       Red
// [1;31m       Bold Red
// [0;32m       Green
// [1;32m       Bold Green
// [0;33m       Yellow
// [01;33m      Bold Yellow
// [0;34m       Blue
// [1;34m       Bold Blue
// [0;35m       Magenta
// [1;35m       Bold Magenta
// [0;36m       Cyan
// [1;36m       Bold Cyan
// [0m  Reset

#include<iostream>
#include<bits/stdc++.h>
using namespace std;

string cipher =
"wsamiepjoysgtmeyipbya.Paxgniphayy,meysywahgmewh
unordered_map<char,char> sbstn; //substitution
hash map

void show_message(){
    char k;
    for(int i=0; i<cipher.length();i++){
        char c = tolower(cipher[i]);
        if(sbstn.find(c)!=sbstn.end()){
            printf("\033[1;31m");
            if(cipher[i]<'a' and cipher[i]>='A')
{
                k = sbstn[c];
                k = toupper(k);
```

```
                cout<<k;
            }
            else printf("%c",sbstn[c]);
            printf("\033[0m");
        }
        else {
            if(cipher[i]<'a' and cipher[i]>='A')
{
                k = toupper(c);
                cout<<k;
            }
            cout<<c;
        }
    }
    cout<<endl;
}


int main(){
    unordered_map<char,double> hash;
    multimap<double,char,greater<double>> freq;
//stores frequency
    int totl_letters=0;
    for(int i=0; i<cipher.length();i++){
        char c = tolower(cipher[i]);
        if(isalpha(c)){
            totl_letters++;
            if(hash.find(c)==hash.end()) hash[c]
= 1;
            else hash[c]++;
        }
    }
    for(auto it=hash.begin();it!=hash.end();
it++){
        it->second =
(it->second/totl_letters)*100;
        // freq[it->second] = it->first;
        freq.insert({it->second,it->first});
    }
    cout<<"Frequency Analysis:\n";
    for(auto it=freq.begin();it!=freq.end();
it++){
        cout<<it->second<<" - "<<it->first<<"%"
<<endl;
    }

        //step 1 . y is highest frequency
replace it with 'e'. 'm' and 'e' are substiuted
for 'the'.
    cout<<"\nStep 1\n"<<"-----------------\n";
    sbstn['y']= 'e';
```

```
74      sbstn['m'] = 't';
75      sbstn['e'] = 'h';
76
77      show_message();
78      cipher = "wsamiepjoysgt mey ipbya.Paxgniphay
iepjoys.Agjygt mey kpmysiepjoysavwkkoyjgsywhmysy
iguynayutgsmewajyaaprywapawjfkyanoamwmnmwghiwfey
Mey fpaavgsuwamxSrN03uwddvwmegnm mey dngmya.Mewa
79      cout<<"\nAfter adding space around
'the':\n";
80      show_message();
81
82          //step 2. In english language , in
middle sentence if 'the' is present . some
preposition
83          //        like 'on','of','at','in' etc .
follows 'the'. In given cipher 'gt' follows
'the'.
84          //        By brute force , we find out
'o' -> 'g' and 'f' -> 't'.
85      sbstn['g'] = 'o';
86      sbstn['t'] = 'f';
87
88          // step 2.2 by looking at last two word
'thw awathef' .break it by 'the'. we get 'thw
awa the f'. w can not be 'e'. Since a
preposition precedes the word "the" we broke it
into 'thwa wa the f'.
89          //by looking at the word generated by
th_ _ (length 4). we find out 'w' -> i and 'a'
-> 's' as it satisfyies with the preopsition
-"is" precedig the word "the".
90      cout<<"\nStep 2\n"<<"-----------------\n";
91      sbstn['w'] = 'i';
92      sbstn['a'] = 's';
93      // This leads to the generation of the word
"this" from the cipher text "mewa".
94      cipher = "wsamiepjoys gt mey ipbya.Paxgnipha
hgmewhr gt whmysyamwh mey iepjoys.Agjy gt mey
kpmysiepjoysavwkkoyjgsywhmysyamwhrmeph mewa ghy!
iguynayutgs mewa jyaapry wa
pawjfkyanoamwmnmwghiwfeyswhvewieuwrwmaepbyoyyhae
Mey fpaavgsu wa mxSrN03uwddvwmegnm mey dngmya. M
95      cout<<"\nAfter adding space around 'of',
'this' and 'is':\n";
96      show_message();
97
98      // step 3.1  We have 'i_te_esti_ the' in our
message. Since a preopsition precedes the word
'the' "ih" seems to be "in". so 'n' -> 'h'
99      sbstn['h'] = 'n';
```

```
100
101     // step 3.2 by looking at 'of inte_est' ,we
    can say the word is 'interest'. 'r' -> 's'.
102     sbstn['s'] = 'r';
103     cipher = "wsamiepjoys gt mey ipbya.Paxgnipha
    hgmewhr gt whmysyam wh mey iepjoys.Agjy gt mey
    kpmysiepjoysavwkkoyjgsy whmysyam whrmeph mewa gh
    iguynayutgs mewa jyaapry wa
    pawjfkyanoamwmnmwghiwfeyswhvewieuwrwmaepbyoyyhae
    Mey fpaavgsu wa mxSrN03uwddvwmegnm mey dngmya. M
104     cout<<"\nStep 3\n"<<"-----------------\n";
105     cout<<"\nAfter adding space around
    'interest':\n";
106     show_message();
107
108     //step 4. by looking at 'the re is not hin_
    of interest', the actual sentence is read by us
    as 'there is nothinr of interest'. 'nothinr'
    looks like 'nothing', So, 'g' -> 'r'.
109     sbstn['r'] = 'g';
110     // Making some more space correction.
111     cipher = "wsamiepjoys gt mey ipbya.Paxgnipha
    hgmewhr gt whmysyam wh mey iepjoys.Agjy gt mey
    kpmysiepjoysavwkkoyjgsy whmysyamwhr meph mewa gh
    tgs mewa jyaapry wa
    pawjfkyanoamwmnmwghiwfeyswhvewieuwrwmaepbyoyyhae
    Mey fpaavgsu wa mxSrN03uwddvwmegnm mey dngmya. M
112     cout<<"\nStep 4\n"<<"-----------------\n";
113     cout<<"\nAfter correcting the place of
    spaces :\n";
114     show_message();
115
116     //step 5.1 by looking at 'interesting th_n
    this one!'. the actual sentence is read by us as
    'interesting than this one!'. 'thpn' looks like
    'than'. 'a' -> 'p'.
117     sbstn['p'] = 'a';
118     //step 5.2 by looking at 'so_e of the
    _ater". the actual sentence is read by us as
    'some of the later'. soje looks 'some' and
    'kater' like 'later'. 'j' -> 'm' and 'k' -> 'l'.
119     sbstn['j'] = 'm';
120     sbstn['k'] = 'l';
121     //step 5.3 by looking at '_e more
    interesting' it looks like 'be more
    interesting'.  'o' -> 'b'.
122     sbstn['o'] = 'b';
123
124     cipher = "wsamiepjoys gt mey
    ipbya.Paxgniphayy, meysy wa hgmewhr gt whmysyam
    wh mey iepjoys.Agjy gt mey kpmys iepjoysavwkk oy
```

```
    jgsy whmysyamwhr meph mewa ghy! Mey iguynayu tgs
    mewa jyaapry wa
    pawjfkyanoamwmnmwghiwfeyswhvewieuwr wma epby
    oyyh aewtmyuox8fkpiya. Mey fpaavgsu wa
    mxSrN03uwddvwmegnm mey dngmya. Mewa wa mey t";
125     cout<<"\nStep 5\n"<<"-----------------\n";
126     cout<<"\nAfter substituting p,j,k,o, and b
    and positioning spaces :\n";
127     show_message();
128
129     //step 6.1 by looking at 'ha_e been sh ifte
    u'. it looks like 'have been shifted' . 'b' ->
    'v' and 'u' -> 'd'.
130     sbstn['b'] = 'v';
131     sbstn['u'] = 'd';
132     //step 6.2. by looking at '_hamber of the
    _aves '. It looks like 'chamber of the caves'.
    'i' -> 'c'.
133     sbstn['i'] = 'c';
134     //step 6.3 by looking at 'chamers _ill be
    more' It looks like 'chamber will be more'. 'v'
    -> 'w'.
135     sbstn['v'] = 'w';
136
137     cipher = "wsam iepjoys gt mey ipbya. Paxgn
    iph ayy, meysy wa hgmewhr gt whmysyam wh mey
    iepjoys. Agjy gt mey kpmys iepjoysa vwkk oy jgsy
    whmysyamwhr meph mewa ghy! Mey iguy nayu tgs
    mewa jyaapry wa pawjfkyanoamwmnmwghiwfeys wh
    vewie uwrwma epby oyyh aewtmyu ox8fkpiya. Mey
    fpaavgsu wa mxSrN03uwddvwmegnm mey dngmya. Mewa
    wa mey t";
138     cout<<"\nStep 6\n"<<"-----------------\n";
139     cout<<"\nAfter substituting b,u,i,v and
    positioning spaces :\n";
140     show_message();
141
142     //step 7.1 - by looking at 'the code _sed
    for this message'. It looks like 'the code used
    for this message'. Hence 'n' -> 'u'.
143     sbstn['n'] = 'u';
144     //step 7.2 - by looking at 'as_o_ can see'.
    It looks like 'as you can see'. 'x' -> 'y'.
145     sbstn['x'] = 'y';
146     //step 7.3 - by looking at 'flaces' and
    'duotes' individually . '_assword' is 'password'
    and '_uotes' is 'quotes'. So, 'f' -> 'p' and 'd'
    -> 'q'.
147     sbstn['f'] = 'p';
148     sbstn['d'] = 'q';
149     cipher = "wsam iepjoys gt mey ipbya. Pa xgn
```

```
iph ayy, meysy wa hgmewhr gt whmysyam wh mey iepjoys. Agjy gt mey kpmys iepjoysa vwkk oy jgsy whmysyamwhr meph mewa ghy! Mey iguy nayu tgs mewa jyaapry wa p awjfky anoamwmnmwgh iwfeys wh vewie uwrwma epby oyyh aewtmyu ox 8 fkpiya. Mey fpaavgsu wa mxSrN03uwdd vwmegnm mey dngmya. Mewa wa mey t";
    cout<<"\nStep 7\n"<<"-----------------\n";
    cout<<"\nAfter substituting n,x,f,d and positioning spaces :\n";
    show_message();

    // Step 8 - Last incomplete sentence "This is the f" seems that it is the part of the 1st sentence. So rearranging it.
    cipher = "Mewa wa mey twsam iepjoys gt mey ipbya. Pa xgn iph ayy, meysy wa hgmewhr gt whmysyam wh mey iepjoys. Agjy gt mey kpmys iepjoysa vwkk oy jgsy whmysyamwhr meph mewa ghy! Mey iguy nayu tgs mewa jyaapry wa p awjfky anoamwmnmwgh iwfeys wh vewie uwrwma epby oyyh aewtmyu ox 8 fkpiya. Mey fpaavgsu wa mxSrN03uwdd vwmegnm mey dngmya.";
    cout<<"\nStep 8\n"<<"-----------------\n";
    cout<<"\nDecrypted Plain Text :\n";
    show_message();
//so final decrypted text is
/*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 shiftedby 8 places. the password is tyrgu03diqq without the quotes. */




    return 0;
}
```

Assignment 1

● UNGRADED

GROUP

ROHIT RAJ
MOHIT KUMAR
ADITYA JAIN
View or edit group

TOTAL POINTS

- / 50 pts

QUESTION 1

Commands 5 pts

QUESTION 2

Cryptosystem 5 pts

QUESTION 3

Analysis 25 pts

QUESTION 4

Mapping 10 pts

QUESTION 5

Password 5 pts

QUESTION 6

Codes 0 pts