Q1 Commands
5 Points

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

["climb", "read", "enter", "read"]

Q2 Cryptosystem
5 Points

What cryptosystem was used at this level?

Substitution Cipher, ROT Encryption

Q3 Analysis
25 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.

As we only had access to the ciphertext, we decided to use frequency analysis and bruteforce attack on the ciphertext. First of all we attacked the one and two lettered words as explained in the class. For example there is high chances that "is" comes after "This" and a single lettered word can be "a" or "I". After that we noticed a lot of "m" in the beginning of the sentences which matches the real life frequency of the letter "t" ("the", "they", "this", "that"). After mapping a few characters, we had ciphertext broken a bit which looked like the following:

"""
This is thy tisst ihajbys ot thy iavys. As xou iah syy, thysy is

hothihr ot ihtysyst ih thy ihajbys. Sojy ot thy katys ihajbyss wikk by josy ihtysystihr thah this ohy! Thy iody usyd tos this jyssary is a sijfky substitutioh iifhys ih whiih dirits havy byyh shittyd bx 8 fkaiys. Thy fasswosd is "txSrU03didd" without thy duotys.
"""

We got strong hints from the partially broken ciphertext above. Using it, we attacked more such texts which started to make meaning.

After many such iterations, finally we were able to make sense of all the words.

Only task left was to decipher the digits, for which we used a hint from the almost broken ciphertext:
"The code used for this message is a simple substitution cipher in which digits have been shifted by 8 places."

This '8' was also a ciphertext left for us to decipher. We supposed the correct shifting was x.
Hence when x shifted by x placed must be equal to 8.
(x+x)mod 10 = 8
which gives x=4 or x=9

Now we bruteforce attacked with both the possibilities and got the correct password with x=4.
in cipher text we had digits 8, 0 and 3 which were made by shifting 4, 6 and 9 by 4 places.

The final plaintext we received was:
"""
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.
"""

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 = [A-Z, a-z, 0-9]
Ciphertext space = [A-Z, a-z, 0-9]

Note-
Capital letter in Plaintext remains capital in the ciphertext.
Small letter in Plaintext remains small in the ciphertext.

The mapping from plaintext to ciphertext is: [a->p, b->o, c->i, d->u, e->y, f->t, g->r, h->e, i->w, l->k, m->j, n->h, o->g, p->f, q->d, r->s, s->a, t->m, u->n, v->b, w->v, y->x, 0->4, 1->5, 2->6, 3->7, 4->8, 5->9, 6->0, 7->1, 8->2, 9->3]

It is observed that j,k,x,z have frequency=0 in the plaintext and c,l,q,z have frequency=0 in ciphertext and hence not been mapped.

Q5 Password
5 Points

What is the final command used to clear this level?

tyRgU69diqq

Q6 Codes
0 Points

Upload any code that you have used to solve this level

5.cpp Download

```
#pragma GCC optimize("Ofast,unroll-loops")
#pragma GCC target("avx,avx2,fma")
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

```

```
#ifndef ONLINE_JUDGE
#define dbg(x) cerr<<#x<<" "; _print(x); cerr<<'\n';
#include "MyDebug.hpp"
#else
#define dbg(...)
#define TIME()
#endif

typedef long long int ll;
typedef long double ld;

typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update> ordered_set;
typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag, tree_order_statistics_node_update> ordered_multiset;
typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll> >, rb_tree_tag, tree_order_statistics_node_update> ordered_pair_set;
#define forder(x) find_by_order(x)
#define okey(x) order_of_key(x)

typedef pair<ll,ll> pii;
typedef vector<ll> vi;
typedef vector<pii> vpii;
typedef vector<vector<ll> > vvi;
typedef vector<vector<pii> > vvpii;

#define all(v) (v).begin(), (v).end()

#define cases() int test_case; cin >> test_case; while(test_case--)
#define fastio ios_base::sync_with_stdio(false); cin.tie(NULL); cout.tie(NULL);

void solve(){
    map<char, char> mp;
    string s="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.";
    // This mp is the reverse map from ciphertext to plaintext
    mp['f']='p';
    mp['p']='a';
    mp['a']='s';
    mp['v']='w';
    mp['g']='o';
    mp['s']='r';
```

```
    mp['u']='d';
    mp['m']='t';
    mp['x']='y';
    mp['e']='h';
    mp['y']='e';
    mp['w']='i';
    mp['t']='f';
    mp['i']='c';
    mp['r']='g';
    mp['n']='u';
    mp['d']='q';
    mp['b']='v';
    mp['h']='n';
    mp['j']='m';
    mp['o']='b';
    mp['k']='l';

    vector<pair<char, char> > v;
    for(auto i:mp){
        v.push_back({i.second, i.first});
    }

    sort(all(v));
    for(auto i:v){
        cout<<i.first<<"->"<<i.second<<", ";
    }cout<<'\n'<<"Unmapped: \n";

    for(int i=0;i<26;i++){
        char c=(char)('a'+i);
        bool found=false;
        for(auto j:mp){
            if(j.second==c){
                found=true;
            }
        }
        if(!found){
            cout<<c<<" ";
        }
    }cout<<'\n';

    for(int i=0;i<26;i++){
        if(mp.find((char)('a'+i))==mp.end()){
            cout<<(char)('a'+i)<<" ";
        }
    }cout<<'\n';

    for(int i=0;i<10;i++){
        mp[(char)('0'+i)]=(char)('0'+(i+6)%10);
    }
    cout<<"Final Answer: \n";
    for(auto i:s){
        if(i<='Z' && i>='A'){
```

```
            if(mp.find(i+('a'-'A'))!=mp.end()){
                cout<<(char)(mp[i+('a'-'A')]+('A'-'a'));
            }else{
                cout<<i;
            }
            continue;
        }
        if(mp.find(i)!=mp.end()){
            cout<<mp[i];
        }else{
            cout<<i;
        }
    }cout<<'\n';
}

int main(){
    fastio

    #ifndef ONLINE_JUDGE
    // freopen("input.txt", "r", stdin);
    // freopen("output.txt", "w", stdout);
    // freopen("error.txt", "w", stderr);
    #endif
    // cases()
    solve();

    // TIME();
        return 0;
}
```

Q7 Team Name

0 Points

team_13

Assignment 1

● Graded

Group

SUKET RAJ
UTTAM KUMAR
SOLANKI KEVALKUMAR BALVANTBHAI
View or edit group

Total Points

31 / 50 pts

Question 1

Commands **5** / 5 pts

Question 2

Cryptosystem **3** / 5 pts

Question 3

Analysis **13** / 25 pts

Question 4

Mapping **5** / 10 pts

Question 5

Password **5** / 5 pts

Question 6

Codes **0** / 0 pts

Question 7

Team Name **0** / 0 pts