Assignment 2

● Graded

Group

KUNAL SAINI
KUSHAL MAHESHWARI
SHARATH KUMAR V

View or edit group

Total Points

50 / 65 pts

Question 1

Commands

10 / 10 pts

✓ **+ 10 pts** commands: 1. read back go 2. go back read , 3. go go read (answer such as for ciphertext: read and for key : go , is also correct)

+ 10 pts command : read, full marks only if the student tried to solve the assignment correctly without the provided key length

Question 2

Cryptosystem

10 / 10 pts

+ 0 pts Incorrect

✓ **+ 10 pts** Correct Cryptosystem.: Vigenere Cipher with Key jcjcffccb.

– 2 pts Wrong Key

Question 3

Analysis

Resolved **10** / 20 pts

✓ **+ 5 pts** Proper mention about if they tried shift cipher, mono alphabetic substitution cipher etc. before concluding its poly alphabetic substitution cipher.

✓ **+ 5 pts** Mention about key 9 2 9 2 5 5 2 2 2 1(jcjcffcccb), Key length 10 and working with key length 10 anywhere in Q4 or Q3

+ 5 pts Assigning them 9 2 9 2 5 5 2 2 2 1 as A to 0, B to 1, to get the key JCJCFFCCCB anywhere in Q4 or Q3 / Doing frequency analysis to for the mapping and therefore finding the key anywhere in Q4 or Q3

+ 5 pts Use of the Kasiski test / Index of Coincidence/Repetition of same blocks to figure out the cryptosystem anywhere in Q3 and Q4

+ 0 pts Incorrect

Regrade Request **Submitted on: Feb 22**

The key utilized in the decryption process was represented numerically, rather than in the English alphabet. This was accomplished by converting the cipher text into a sequence of numbers, as described in the code.
To decrypt the message, we employed the following formulas in C++: (ch - 'a' - pattern[i] + 26) mod 26 + 'a' and (ch - 'A' - pattern[i] + 26) mod 26 + 'A'. These formulas were used to define the character 'a' as 0, 'b' as 1, and so forth for lowercase letters, while similar equations were used to represent capital letters. Thus, it was not necessary to have the key in alphabetical form.
In terms of cryptanalysis, the Kasiski test was deemed unnecessary due to the helpful hint provided. We were able to crack the cipher using polyalphabetic substitution, which obviated the need for further analysis using the Kasiski test.

The grading is just fine based on the answer provided in Q3 and Q4

Reviewed on: Feb 22

Question 4

Decryption Algorithm

Resolved **10** / 15 pts

✓ **+ 5 pts** Mentioning removal of spaces/punctuation etc., or mentioning mapping of them is fixed and mentioning about "spaces" while calculating the distance of blocks anywhere in Q3 and Q4

+ 5 pts mentioning (plaintext alphabet + key) mod 26 = cipher text alphabet or (cipher text - key) mod 26 = plaintext anywhere in Q3 or Q4

✓ **+ 5 pts** Mentioning breaking the ciphertext into 10-size blocks and giving a detailed description of decoding or providing codes to get the plaintext anywhere in Q3 and Q4

+ 0 pts Incorrect

+ 5 pts Correct answer found but explanation is not found.

Regrade Request **Submitted on: Feb 22**

In Q.4 we have clearly mentioned that block only contains alphabets inside the cipher text which implies that there is no spaces, and other characters are considered in the block.
Another regrade point is that - We have written [(cipher text - key + 26) mod 26 = plain text] through text itself, "For ith element of each block shift it cyclically by -k[i](ith element of the key) to obtain the plain text", this sentence in out statement means the same.
We have also mentioned the size of block of length 10 that means the same as for breaking the cipher text in blocks of size 10.

For first case : marks are added. For second case , its not implying the case what happens if the cycle goes beyond 26. hence marks cant be given. +10 added

Reviewed on: Feb 22

Question 5

Password

10 / 10 pts

+ 0 pts Incorrect

✓ **+ 10 pts** Correct

Question 6

Codes

0 / 0 pts

✓ **+ 0 pts** Correct

Question 7

Team Name

0 / 0 pts

✓ **+ 0 pts** Correct

+ 0 pts Incorrect

Q1 Commands

10 Points

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

go
read

Q2 Cryptosystem

10 Points

What cryptosystem was used in this level?

Multiple Substitution Cipher(Polyalphabetic substitution cipher)

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.

When we use the command to "go" to check the funny pattern written in the distance boulder, we notice that it says "Bow, and then slowly look up. Count the number of lines in horizontal dimension". Which means we need to count the lines from the bottom to the top of the image. By doing this we get an array key=[9,2,9,2,5,5,2,2,2,1], this must be the key to the cipher. It cannot be a permutation cipher as there are repeated numbers. So this must be a multiple substitution cipher and the numbers must indicate the amount by which the letters are shifted.

Q4 Decryption Algorithm

15 Points

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

Algorithm: Since the key is of size 10, make a block of size 10 containing alphabets in the cipher text. For ith element of each block shift it cyclically by the -k[i](ith element of key) to obtain the plain text.

Plain text: 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

▾ test.cpp ⤓ Download

```cpp
#include<iostream>
#include<vector>
using namespace std;

int main(){

    vector<int> key{9,2,9,2,5,5,2,2,2,1};
    string cipher = "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.";
    int k=0;

    for(int i=0; i<cipher.size(); i++)
    {
        if('a'<=cipher[i] && cipher[i]<='z')
        {
            cipher[i] = (cipher[i]-'a'-key[k]+26)%26 + 'a';
            k = (k+1)%10;
        }
        else if('A'<=cipher[i] && cipher[i]<='Z')
        {
            cipher[i] = (cipher[i]-'A'-key[k]+26)%26 + 'A';
            k = (k+1)%10;
        }
    }

    cout << cipher << endl;

    return 0;

}
```

Q7 Team Name

0 Points

hardwired