Q1 Team name

0 Points

Cipherberg

Q2 Commands

10 Points

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

go, back, read

Q3 Cryptosystem

10 Points

What cryptosystem was used in this level?

Playfair cipher (a Symmetric Key Cryptosystem) and Morse code

Q4 Analysis

20 Points

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

Tools Used: International Morse Code Chart (https://en.wikipedia.org/wiki/Morse_code), python.
Observations:

1. Starting from level 2, we entered 'go' command to get to the patterns on one of the boulders. We could find some patterns in the form of dots and dashes which looked like "Morse Code". To decrypt it, we used the International Morse Code Chart and

interestingly, we found that the Morse Code converts to "SECURITY". The word "SECURITY" may be the key of cipher text.

2. The words of the cave man "Believe in yourself and PLAY FAIR", strongly indicate that somewhere ahead in this level, we may have to use playfair cipher to decrypt the ciphertext as his last words are "PLAY FAIR".

3. To reach the cipher text, we go back using "back" command and enter "read" command to read the panel.

4. We first tried performing frequency analysis on the ciphertext, however the obtained frequencies were not consistent with the standard English alphabet frequency table which very likely means that the original text was not encrypted using Substitution cipher and some other encryption technique is used.

5. We then start decrypting it, assuming "Playfair" cipher with "SECURITY" as the key. We have another reason to believe that the cryptosystem used is playfair cipher because the ciphertext does not contain the letter 'J'.

6. The decrypted text obtained had some extra 'X' like "OUT X THE", "MAY X YOU", "WIL X L" and "NE X ED". These extra "X" are removed to get the original message text. Reason: While encrypting the message with playfair cipher, whenever both letters of the bigram are same, an "X" is added after the first letter. We need to remove these extra "X" from the deciphered text to get the original message.

7. After decryption, one word in the text is "IOY" which doesn't make sense as there is no such word in English language. This word is possibly "JOY" because while encryption all the "J" in message are replaced with "I", if the table(5x5 grid) doesn't include "J". Therefore, in the decrypted text, we need to replace "I" with "J" wherever necessary.

8. The special characters(", _ and .) and whitespaces are likely to unchanged.

Q5 Decryption algorithm

15 Points

Briefly describe the decryption algorithm used. Also, mention the plaintext you deciphered. (Use less than 250 words)

For decrypting the ciphertext, we use the following algorithm:
1. We make a 5 x 5 key square grid of alphabets. Each of the 25 alphabets to be placed in the grid must be unique and since the table can hold only 25 unique alphabets, one letter of the alphabet (usually J) is omitted from the table.

2. The first n (=len(keyword)) characters of the table are that of keyword in the same order and the rest (25-n) are remaining English alphabets, lexicographically. Therefore, to obtain the key table we first place the letters of the key(SECURITY) as the initial alphabets and then fill all the remaining alphabets from A to Z (except J). The key table obtained is:
S, E, C, U, R
I, T, Y, A, B
D, F, G, H, K,
L, M, N, O, P
Q, V, W, X, Z

3. We strip the special characters(" , _ , .) and whitespaces from the ciphertext.

4. We split the ciphertext into pairs of two letters called bigrams. If there is an odd number of letters, a Z is added to the last letter. In our case, the cipher text had even number of letters and there wasn't a need to add 'Z'.

5. For each bigram, we follow the following rules for decryption:
i) If both the letters are in the same row, we take the letter to the left of each one. We wrap around to the rightmost position if a letter in the bigram is at the leftmost position.
ii) If both the letters are in the same column, we take the letter above each one, wrapping around to the bottom if a letter in the bigram is at the topmost position .
iii) If neither of the above rules is true, we form a rectangle with the two letters and take the letters on the horizontal opposite corner

of the rectangle.

Deciphered plaintext (With whitespaces and special characters at their original place): "BE WARY OF THE NEXT CHAMBER, THERE IS VERY LITTLE IOY THERE. SPEAK OUT XTHE PASSWORD "OPEN_SESAME" TO GO THROUGH. MAY XYOU HAVE THE STRENGTH FOR THE NEXT CHAMBER. TO FIND THE EXIT YOU FIRST WILXL NEXED TO UTTER MAGIC WORDS THERE."

After removing the extraneous "X" from the deciphered text and replacing "I" with "J" wherever necessary, we obtain the final decrypted message/plaintext as: "BE WARY OF THE NEXT CHAMBER, THERE IS VERY LITTLE JOY THERE. SPEAK OUT THE PASSWORD "OPEN_SESAME" 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."

Q6 Password

10 Points

What was the final command used to clear this level?

OPEN_SESAME

Q7 Code

0 Points

Upload any code that you have used to solve this level.

▾ morse.ipynb Download

In [ ]:

```
morsetoEnglish = {
                '.-': 'A',
                 '-...': 'B',
                 '-.-.': 'C',
                 '-..': 'D',
                 '.': 'E',
                 '..-.': 'F',
                 '--.': 'G',
                 '....': 'H',
                 '..': 'I',
                 '.---': 'J',
         '-.-': 'K',
         '.-..': 'L',
         '--': 'M',
         '-.': 'N',
         '---': 'O',
         '.--.': 'P',
         '--.-': 'Q',
         '.-.': 'R',
         '...': 'S',
         '-': 'T',
         '..-': 'U',
         '...-': 'V',
         '.--': 'W',
         '-..-': 'X',
         '-.--': 'Y',
         '--..': 'Z',
         '/': ' ',
         '.----': '1',
         '..---': '2',
         '...--': '3',
         '....-': '4',
         '.....': '5',
         '-....': '6',
         '--...': '7',
         '---..': '8',
         '----.': '9',
         '-----': '0',
         '.-.-.-': '.',
         '--..--': ',',
         '---...': ':',
         '..--..': '?',
         '.----.': "'",
         '-....-': '-',
         '-..-.': '/',
         '.--.-.': '@',
         '-...-': '='
        }


morseCode = "... . -.-. ..- .-. .. - -.--"


def decrypt(morse):
    spaceSeparated = morse.split(' ')
    decryptedMessage = ''
    for char in spaceSeparated:
        decryptedMessage +=
morseToEnglish[char]
    return decryptedMessage

print(decodeMorse(morseCode))
```

Assignment 2

● GRADED

GROUP

SHRUTI SHARMA
DEEKSHA ARORA
SAMBHRANT MAURYA

✎ View or edit group

TOTAL POINTS

65 / 65 pts

QUESTION 1	Team name	**0** / 0 pts
QUESTION 2	Commands	**10** / 10 pts
QUESTION 3	Cryptosystem	**10** / 10 pts
QUESTION 4	Analysis	**20** / 20 pts
QUESTION 5	Decryption algorithm	**15** / 15 pts
QUESTION 6	Password	**10** / 10 pts
QUESTION 7	Code	**0** / 0 pts