Experiment 4

Shashwat Shah
60004220126
TYBtech Comps B

Aim : Study and implement simple columnar transpositional cipher

Theory : Given a plaintext message and a numeric key, cipher de-cipher the given text using columnar transposition cipher. It is a form of transposition just like rail fence cipher. It involves writing the plaintext out in rows, then reading the cipher text off in columns one by one.

Encryption - In a transposition cipher the order of the alphabet is re-arranged to obtain the cipher text.

1) The message is written out in rows of a fixed length and then read out again column by column and the columns are chosen in some scrambled order.

2) Width of the rows and the permutation of the columns are usually defined by a keyword

3) For Eg, the word HACK is of length 4 (so the rows are of length 4), and the permutation is defined by the alphabetical order of the letters in the keyword. In this case, the order would be '3124'

4) Any spare space are filled with nulls or left blank or placed by a * character (Eg : _)

5) Finally, the message is read of in column in the order specified by keyword.

Print character of columns 1,2,3,4.

Encrypted text - ekefgsgsrekoe-

H	A	C	K
3	1	2	4
G	e	e	k
S	_	f	o
r	_	g	e
e	k	s	_

Decryption : To decipher it, the recepient has to work out the column length by dividing the message length by key length.
Then, write the message out in columns again, then re-order the columns by reforming the keyword.

Conclusion : It is simple and efficient encryption method that has been widely used in ~~narrow~~ various applications, including data protection and military communication, etc.
Hence, we studied and implemented columnar transposition.

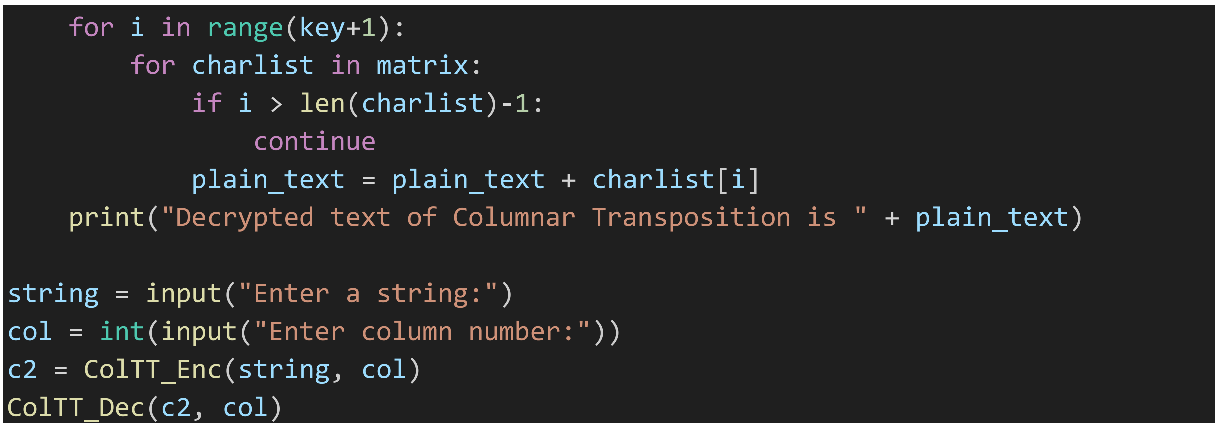
EXPERIMENT 4

Shashwat Shah
TYBtech Comps B
C22
60004220126

AIM: Study and Implement Simple Columnar Transposition Cipher.

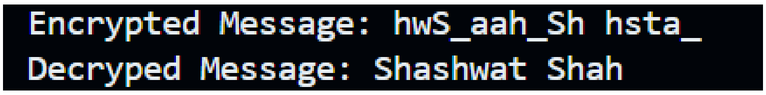
CODE:

```
def ColTT_Enc(plain_text, key):
    matrix = []
    for i in range(key):
        matrix.append([])
    for i in range(len(plain_text)):
        matrix[i % key].append(plain_text[i])
    for i in matrix:
        print(i)
    cypher_text = ''
    for i in matrix:
        for char in i:
            cypher_text += char
    print("Cipher text of Columnar Transposition is " + cypher_text)
    return cypher_text


def ColTT_Dec(cypher_text, key):
    matrix = []
    for i in range(key):
        matrix.append([])
    count = int(len(cypher_text)/key)
    length = 0
    extra = int(len(cypher_text) % key)
    for charlist in matrix:
        for j in range(count):
            charlist.append(cypher_text[length])
            length = length+1
        if (extra != 0):
            charlist.append(cypher_text[length])
            length = length+1
            extra = extra-1
    for i in matrix:
        print(i)
    plain_text = ''
```



```
    for i in range(key+1):
        for charlist in matrix:
            if i > len(charlist)-1:
                continue
            plain_text = plain_text + charlist[i]
    print("Decrypted text of Columnar Transposition is " + plain_text)

string = input("Enter a string:")
col = int(input("Enter column number:"))
c2 = ColTT_Enc(string, col)
ColTT_Dec(c2, col)
```

OUTPUT:



```
Encrypted Message: hwS_aah_Sh hsta_
Decryped Message: Shashwat Shah
```