

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
INS Practical 5

Aim : Alice wants to send some confidential information to Bob over a secure network. Provide encryption through Hill Cipher Method for message "Palladium Mall" and Key is "SAVE" (A=1,B=2...). Also decrypt using same.

Code :

```
import numpy as np
import YSL_io

def transpose(matrix):
    matrix[0, 0], matrix[1, 1] = matrix[1, 1], matrix[0, 0]
    matrix[0, 1] *= -1
    matrix[1, 0] *= -1
    return matrix

def inverse_modulo(det_mod, mod=26):
    k_inv = 1
    while (det_mod * k_inv) % mod != 1:
        k_inv += 1
    return k_inv

def encrypt(plaintext, key_matrix):
    ciphertext_list = []

    for char_pair in plaintext:
        t = np.zeros((2, 1), dtype=np.int64)
        t[0, 0] = ord(char_pair[0]) - 96
        t[1, 0] = ord(char_pair[1]) - 96
        cipher = np.dot(key_matrix, t) % 26
        ciphertext_list.append(chr(cipher[0, 0] + 96) + chr(cipher[1, 0] +
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
INS Practical 5

```
96))

    return ciphertext_list

def decrypt(ciphertext_list, key_matrix):
    det_mod = (
        key_matrix[0, 0] * key_matrix[1, 1] - key_matrix[1, 0] *
key_matrix[0, 1]
    ) % 26
    k_inv = inverse_modulo(det_mod)
    adj_key_matrix = transpose(key_matrix.copy())
    k_inv_matrix = (adj_key_matrix % 26) * k_inv % 26

    plaintext_list = []
    for char_pair in ciphertext_list:
        t = np.zeros((2, 1), dtype=np.int64)
        t[0, 0] = ord(char_pair[0]) - 96
        t[1, 0] = ord(char_pair[1]) - 96
        decipher = np.dot(k_inv_matrix, t) % 26
        plaintext_list.append(chr(decipher[0, 0] + 96) + chr(decipher[1, 0]
+ 96))

    return plaintext_list

plaintext = YSL_io.inputGRN("\nEnter the plain text : ")
key = YSL_io.inputMGNTA("\nEnter the key : ")

if len(plaintext) % 2 != 0:
    plaintext += "x"

temp_key_matrix = np.array(list(key)).reshape((2, 2))
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
INS Practical 5

```
key_matrix = np.zeros((2, 2), dtype=np.int64)

for i in range(2):
    for j in range(2):
        key_matrix[i, j] = ord(temp_key_matrix[i, j]) - 96

YSL_io.printBLU("\nChar key matrix : ", end="\n\n")
print(temp_key_matrix)
YSL_io.printRED("\nInteger key matrix : ", end="\n\n")
print(key_matrix)

plaintext_list = [plaintext[i : i + 2] for i in range(0, len(plaintext),
2)]
ciphertext_list = encrypt(plaintext_list, key_matrix)

YSL_io.printORNG("\nCipher : ", end="")
for char_pair in ciphertext_list:
    print(char_pair, end="")

YSL_io.printORNG("\nDecipher : ", end="")
plaintext_list = decrypt(ciphertext_list, key_matrix)
for char_pair in plaintext_list:
    print(char_pair, end="")
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
INS Practical 5

Output :

The screenshot shows a laptop screen with a code editor on the left and a terminal on the right. The code editor contains a Python script for Hill cipher encryption and decryption. The terminal shows the execution of the script, including the input of plain text, key matrix, and the resulting ciphertext and deciphered text.

```
1 import numpy as np
2 import YSL_io
3
4
5 def transpose(matrix):
6     matrix[0, 0], matrix[1, 1] = matrix[1, 1], matrix[0, 0]
7     matrix[0, 1] *= -1
8     matrix[1, 0] *= -1
9     return matrix
10
11
12 def inverse_modulo(det_mod, mod=26):
13     k_inv = 1
14     while (det_mod * k_inv) % mod != 1:
15         k_inv += 1
16     return k_inv
17
18
19 def encrypt(plaintext, key_matrix):
20     ciphertext_list = []
21
22     for char_pair in plaintext:
23         t = np.zeros((2, 1), dtype=np.int64)
24         t[0, 0] = ord(char_pair[0]) - 96
25         t[1, 0] = ord(char_pair[1]) - 96
26         cipher = np.dot(key_matrix, t) % 26
27         ciphertext_list.append(chr(cipher[0, 0] + 96) + chr(cipher[1, 0] + 96))
28
29     return ciphertext_list
30
31
32 def decrypt(ciphertext_list, key_matrix):
33     det_mod = (
34         key_matrix[0, 0] * key_matrix[1, 1] - key_matrix[1, 0] * key_matrix[0,
35         1] % 26
36     )
37     k_inv = inverse_modulo(det_mod)
38     adj_key_matrix = transpose(key_matrix.copy())
39     k_inv_matrix = (adj_key_matrix % 26) * k_inv % 26
```

Terminal Output:

```
yash ~ 12:12 cd Documents/sem6practicals/INS/
yash ~/INS P main ? 12:12 source venv/bin/activate.fish
yash ~/INS P main ? 12:12 python p5.py

Enter the plain text : yashlakhtariya

Enter the key : yash

Char key matrix :

[['y' 'a']
 ['s' 'h']]

Integer key matrix :

[[25 1]
 [19 8]]

Cipher : booiobwmqgxqbo
Decipher : yashlakhtariya
yash ~/INS P main ? 12:13 |
```