NAME: SUNDEEP A   SRN: PES1UG20CS445

ROLL NO:48   SECTION:H

QUESTION:

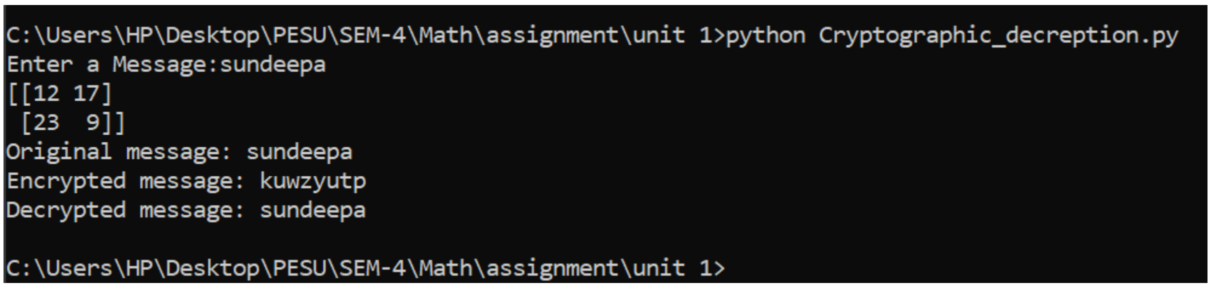
Implementation of Basic Cryptographic Technique to appreciate the Incerse functionality of matrix.

Code:

```python
import numpy as np
from egcd import egcd # pip install egcd
alphabet = "abcdefghijklmnopqrstuvwxyz"
letter_to_index = dict(zip(alphabet, range(len(alphabet))))
index_to_letter = dict(zip(range(len(alphabet)), alphabet))
def matrix_mod_inv(matrix, modulus):
    det = int(np.round(np.linalg.det(matrix)))
    det_inv = egcd(det, modulus)[1] % modulus
    matrix_modulus_inv = (
    det_inv * np.round(det * np.linalg.inv(matrix)).astype(int) % modulus)
    return matrix_modulus_inv
def encrypt(message, K):
    encrypted = ""
    message_in_numbers = []
    for letter in message:
        message_in_numbers.append(letter_to_index[letter])
    split_P = [message_in_numbers[i : i + int(K.shape[0])] for i in range(0,
len(message_in_numbers), int(K.shape[0]))]
    for P in split_P:
        P = np.transpose(np.asarray(P))[:, np.newaxis]
        while P.shape[0] != K.shape[0]:
            P = np.append(P, letter_to_index[" "])[:, np.newaxis]
        numbers = np.dot(K, P) % len(alphabet)
        n = numbers.shape[0]
        for idx in range(n):
            number = int(numbers[idx, 0])
            encrypted += index_to_letter[number]
    return encrypted
def decrypt(cipher, Kinv):
    decrypted = ""
    cipher_in_numbers = []
    for letter in cipher:
        cipher_in_numbers.append(letter_to_index[letter])
    split_C = [cipher_in_numbers[i : i + int(Kinv.shape[0])]
              for i in range(0, len(cipher_in_numbers), int(Kinv.shape[0]))]
    for C in split_C:
        C = np.transpose(np.asarray(C))[:, np.newaxis]
        numbers = np.dot(Kinv, C) % len(alphabet)
        n = numbers.shape[0]
        for idx in range(n):
            number = int(numbers[idx, 0])
            decrypted += index_to_letter[number]
    return decrypted
def main():
    message = input("Enter a Message:")
    K = np.matrix([[3, 3], [1, 4]])
    Kinv = matrix_mod_inv(K, len(alphabet))
    print(Kinv)
    encrypted_message = encrypt(message, K)
    decrypted_message = decrypt(encrypted_message, Kinv)
    print("Original message: " + message)
    print("Encrypted message: " + encrypted_message)
    print("Decrypted message: " + decrypted_message)

main()
```

Screenshot:



```
C:\Users\HP\Desktop\PESU\SEM-4\Math\assignment\unit 1>python Cryptographic_decreption.py
Enter a Message:sundeepa
[[12 17]
 [23  9]]
Original message: sundeepa
Encrypted message: kuwzyutp
Decrypted message: sundeepa

C:\Users\HP\Desktop\PESU\SEM-4\Math\assignment\unit 1>
```