11.Hill Cipher encryption and decryption

Program:

mport numpy as np

def prepare\_message(message, block\_size):

message = message.replace(" ", "").upper()

while len(message) % block\_size != 0:

message += 'X'

return message

def hill\_cipher\_encrypt(message, key\_matrix):

message = prepare\_message(message, 2)

key\_matrix = np.array(key\_matrix)

cipher\_text = ""

block\_size = 2

for i in range(0, len(message), block\_size):

block = message[i:i+block\_size]

block\_vector = np.array([ord(c) - ord('A') for c in block])

encrypted\_vector = np.dot(key\_matrix, block\_vector) % 26

encrypted\_block = ''.join([chr(v + ord('A')) for v in encrypted\_vector])

cipher\_text += encrypted\_block

return cipher\_text

def hill\_cipher\_decrypt(cipher\_text, key\_matrix):

key\_matrix = np.array(key\_matrix)

key\_matrix\_inverse = np.linalg.inv(key\_matrix)

key\_matrix\_inverse = (key\_matrix\_inverse \* np.linalg.det(key\_matrix)).round()

key\_matrix\_inverse = key\_matrix\_inverse.astype(int) % 26

plain\_text = ""

block\_size = 2

for i in range(0, len(cipher\_text), block\_size):

block = cipher\_text[i:i+block\_size]

block\_vector = np.array([ord(c) - ord('A') for c in block])

decrypted\_vector = np.dot(key\_matrix\_inverse, block\_vector) % 26

decrypted\_block = ''.join([chr(v + ord('A')) for v in decrypted\_vector])

plain\_text += decrypted\_block

return plain\_text

key\_matrix = [

[9, 4],

[5, 7]

]

message = "meet me at the usual place at ten rather than eight oclock"

cipher\_text = hill\_cipher\_encrypt(message, key\_matrix)

print("Encrypted message:", cipher\_text)

decrypted\_message = hill\_cipher\_decrypt(cipher\_text, key\_matrix)

print("Decrypted message:", decrypted\_message)

Output:

