4.Polyalphabetic substitution cipher

Program:

def polyalphabetic\_cipher(text, key):

encrypted\_text = ""

key\_length = len(key)

for i in range(len(text)):

char = text[i]

if char.isalpha():

shift = ord(key[i % key\_length].lower()) - ord('a')

if char.isupper():

encrypted\_char = chr(((ord(char) - ord('A') + shift) % 26) + ord('A'))

else:

encrypted\_char = chr(((ord(char) - ord('a') + shift) % 26) + ord('a'))

else:

encrypted\_char = char

encrypted\_text += encrypted\_char

return encrypted\_text

message = input("Enter a message: ")

keyword = input("Enter a keyword: ")

encrypted\_message = polyalphabetic\_cipher(message, keyword)

print("Encrypted message:", encrypted\_message)

Output:

