**COMSATS University Islamabad, Attock campus Department of Computer Science**

**Graded task #02**

**Information Security**

Brekhna Gul

Sp24-bse-051

14-October -2025

Ma’am Ambreen

**Transportation cipher:**

**Code:**

import random

# Split text into chunks

def split\_len(text, size):

    return [text[i:i+size] for i in range(0, len(text), size)]

# Generate random key if user doesn’t give one

def generate\_key(length):

    nums = list(range(1, length + 1))

    random.shuffle(nums)

    return ''.join(str(n) for n in nums)

# Encode function

def encode(key, text):

    if key == "":   # Task 5

        key = generate\_key(len(text))

        print("Generated Key:", key)

    while len(text) % len(key) != 0:   # Task 1 (padding)

        text += "\_"

    order = {int(val): num for num, val in enumerate(key)}

    cipher = ""

    for index in sorted(order.keys()):

        for part in split\_len(text, len(key)):

            try:

                cipher += part[order[index]]

            except:

                pass

    return cipher

# Decode function

def decode(key, cipher):

    order = {int(val): num for num, val in enumerate(key)}

    cols, rows = len(key), len(cipher)//len(key)

    grid = [""] \* rows

    i = 0

    for index in sorted(order.keys()):

        for r in range(rows):

            grid[r] += cipher[i]

            i += 1

    return ''.join(grid).replace("\_", "")

# Menu system

while True:

    print("\n--- Transposition Cipher Menu ---")

    print("1. Encode\n2. Decode\n3. Exit")

    ch = input("Enter choice: ")

    if ch == '1':

        text = input("Enter plaintext: ")

        key = input("Enter key (leave blank for random): ")

        print("Ciphertext:", encode(key, text))

    elif ch == '2':

        cipher = input("Enter ciphertext: ")

        key = input("Enter key used: ")

        print("Plaintext:", decode(key, cipher))

    elif ch == '3':

        print("Exiting...")

        break

    else:

        print("Invalid input!")

**Expected output:**



**Explanation:**

This program performs Transposition Cipher encryption and decryption with added features:

1. Padding – Adds \_ if text isn’t divisible by key length.

2. Decode Function – Reverses the encryption to get original text.

3. Case Handling – Keeps uppercase/lowercase and spaces unchanged.

4. Random Key – Automatically creates a random key if none given.

5. Menu Interface – Lets user choose to encode, decode, or exit.