Name: Faraz Pathan

Rollno: COTB70

'''Write a Java/C/C++/Python program to perform encryption and decryption using the method of

Transposition technique'''

def encryptRailFence(text, key):

    rail = [['\n' for i in range(len(text))]

                for j in range(key)]

    dir\_down = False

    row, col = 0, 0

    for i in range(len(text)):

        if (row == 0) or (row == key - 1):

            dir\_down = not dir\_down

        rail[row][col] = text[i]

        col += 1

        if dir\_down:

            row += 1

        else:

            row -= 1

    result = []

    for i in range(key):

        for j in range(len(text)):

            if rail[i][j] != '\n':

                result.append(rail[i][j])

    return("" . join(result))

def decryptRailFence(cipher, key):

    rail = [['\n' for i in range(len(cipher))]

                for j in range(key)]

    dir\_down = None

    row, col = 0, 0

    for i in range(len(cipher)):

        if row == 0:

            dir\_down = True

        if row == key - 1:

            dir\_down = False

        rail[row][col] = '\*'

        col += 1

        if dir\_down:

            row += 1

        else:

            row -= 1

    index = 0

    for i in range(key):

        for j in range(len(cipher)):

            if ((rail[i][j] == '\*') and

            (index < len(cipher))):

                rail[i][j] = cipher[index]

                index += 1

    result = []

    row, col = 0, 0

    for i in range(len(cipher)):

        if row == 0:

            dir\_down = True

        if row == key-1:

            dir\_down = False

        if (rail[row][col] != '\*'):

            result.append(rail[row][col])

            col += 1

        if dir\_down:

            row += 1

        else:

            row -= 1

    return("".join(result))

if \_\_name\_\_ == "\_\_main\_\_":

    print(encryptRailFence("Computer Science", 3))

    print(decryptRailFence("Cu eoptrSinemecc", 3))

**OUTPUT:**

Cu eoptrSinemecc

Computer Science