CODE:

def encryptRailFence(plain\_text, depth):

    rail = [['\n' for i in range(len(plain\_text))] for j in range(depth)]

    dir\_down = False

    row, col = 0, 0

    for i in range(len(plain\_text)):

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

            dir\_down = not dir\_down

        rail[row][col] = plain\_text[i]

        col += 1

        if dir\_down:

            row += 1

        else:

            row -= 1

    result = []

    for i in range(depth):

        for j in range(len(plain\_text)):

            if rail[i][j] != '\n':

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

    return "" . join(result)

def decryptRailFence(cipher\_text, depth):

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

    dir\_down = None

    row, col = 0, 0

    for i in range(len(cipher\_text)):

        if row == 0:

            dir\_down = True

        if row == depth - 1:

            dir\_down = False

        rail[row][col] = '\*'

        col += 1

        if dir\_down:

            row += 1

        else:

            row -= 1

    index = 0

    for i in range(depth):

        for j in range(len(cipher\_text)):

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

                    (index < len(cipher\_text))):

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

                index += 1

    result = []

    row, col = 0, 0

    for i in range(len(cipher\_text)):

        if row == 0:

            dir\_down = True

        if row == depth - 1:

            dir\_down = False

        if rail[row][col] != '\*':

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

            col += 1

        if dir\_down:

            row += 1

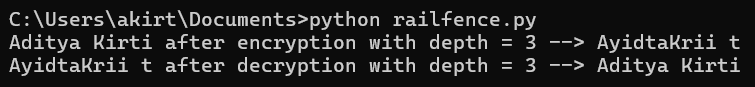
        else:

            row -= 1

    return "".join(result)

print(f'Aditya Kirti after encryption with depth = 3 --> {encryptRailFence('Aditya Kirti', 3)}')

print(f'AyidtaKrii t after decryption with depth = 3 --> {decryptRailFence('AyidtaKrii t', 3)}')

OUTPUT: