

Assignment No 2

```
def encrypt_transposition(plain_text, key):
    cipher_text = [""] * key

    for col in range(key):
        pointer = col
        while pointer < len(plain_text):
            cipher_text[col] += plain_text[pointer]
            pointer += key
    return "".join(cipher_text)

def decrypt_transposition(cipher_text, key):
    num_cols = len(cipher_text) // key
    num_rows = key
    num_shaded_boxes = (num_cols * num_rows) - len(cipher_text)

    plain_text = [""] * num_cols
    col, row = 0, 0

    for symbol in cipher_text:
        plain_text[col] += symbol
        col += 1

        if (col == num_cols) or (col == num_cols - 1 and row >= num_rows -
num_shaded_boxes):
            col = 0
            row += 1
    return "".join(plain_text)

# Example usage
message = "HelloWorld"
key = 4
encrypted_msg = encrypt_transposition(message, key)
print("Encrypted:", encrypted_msg)

decrypted_msg = decrypt_transposition(encrypted_msg, key)
print("Decrypted:", decrypted_msg)
```

Output

Encrypted: HorelWlold
Decrypted: HelloWorld

