Laboratory 4: Implementing Columnar Transposition Cipher in Python

Objective:

Create a Python program that can **encrypt** and **decrypt** messages using the **Columnar Transposition Cipher** technique. This exercise will enhance your understanding of string manipulation, matrix operations, and basic encryption methods in Python.

Background:

The Columnar Transposition Cipher is a type of encryption where:

1. The message is written in rows according to a fixed "key" (number of columns).
2. Characters are read off column-by-column to form the encrypted message.
3. For decryption, the process is reversed to retrieve the original message.

For example:

- **Plaintext**: "Love is not blind."
- **Key**: 8 (number of columns)
- **Encrypted Message**: "Lndoo.vte__bilsi_n"

Task:

Write a Python program that performs the following steps:

1. **Encrypt a Message**: Implement the encryption process by arranging the characters in rows and reading them off column by column.
2. **Decrypt the Message**: Rebuild the original message by reversing the columnar process using the same key.

Instructions

Part 1: Encryption Process

1. **Input the Plaintext and Key**:
   - Prompt the user to enter a message and an integer key (number of columns).
   - Remove any spaces or special characters from the message to create a continuous string.
2. **Arrange Characters in a Grid**:
   - Divide the message into rows based on the key (number of columns). Each row will contain up to key characters.

- If the last row has fewer characters than the key, pad the row with underscores _ to fill all columns.

3. **Read Column by Column**:
   - Start from the first column and move downwards, collecting each character in the column.
   - Continue for each column until the end, skipping any underscores.

4. **Display the Encrypted Message**:
   - Concatenate the collected characters to display the encrypted message.

Part 2: Decryption Process

1. **Calculate Rows and Columns**:
   - Based on the message length and the key, determine the number of rows required for decryption.

2. **Draw the Matrix**:
   - Arrange the ciphertext into columns and rows based on the key.

3. **Fill in Characters**:
   - Place each character from the ciphertext in the appropriate column, skipping any padded characters at the end.

4. **Retrieve Original Message**:
   - Read the characters row by row to recreate the original message.

Example Usage

Input
Copy code

```
Enter the plaintext message: Love is not blind
Enter the key (number of columns): 8
```

Output
Copy code

```
Encrypted Message: Lndoo.vte__bilsi_n
Decrypted Message: Loveisnotblind
```