Implementing Substitution and Transposition Ciphers

Aim: To study and implement the Substitution and Transposition Ciphers

Theory:

Substitution Cipher:
The Substitution Cipher is one of the simplest and oldest methods of encrypting messages. It falls under the category of symmetric key encryption, meaning the same key is used for both encryption and decryption. In a Substitution Cipher, each letter in the plaintext (original message) is replaced by another letter or symbol to create the ciphertext (encrypted message). This method is called substitution because each letter is substituted with another according to a predetermined rule.

Caesar Cipher:
One of the most famous examples of a Substitution Cipher is the Caesar Cipher, named after Julius Caesar, who is believed to have used this method to protect his confidential correspondence. The Caesar Cipher involves shifting each letter in the plaintext by a fixed number of positions in the alphabet.

For example, with a shift of 3, the letter 'A' is substituted with 'D', 'B' with 'E', 'C' with 'F', and so on. This process wraps around the alphabet, so 'X' becomes 'A', 'Y' becomes 'B', and 'Z' becomes 'C'. The shift value is often referred to as the key, and it determines the mapping from plaintext to ciphertext.

Encryption Process:
To encrypt a message using the Caesar Cipher, follow these steps:x

Choose a shift value (key) for the cipher.
Take the plaintext message and, for each letter:

a) Determine its position in the alphabet.
b) Shift the position by the key value.
c) Map the new position back to a letter in the alphabet.
d) Replace the original letter with the mapped letter to obtain the ciphertext.

For example, with a shift of 3, the plaintext "HELLO" would become "KHOOR" in ciphertext.

Decryption Process:
To decrypt a message encrypted with the Caesar Cipher, the recipient needs to know the shift value (key) that was used. The decryption process is the reverse of the encryption process:

Obtain the ciphertext message.
For each letter:

Determine its position in the alphabet.

a) Shift the position back by the key value (subtract the key).
b) Map the new position back to a letter in the alphabet.
c) Replace the original letter with the mapped letter to obtain the plaintext.

Using the same shift of 3, the ciphertext "KHOOR" would be decrypted as "HELLO".

Transposition Cipher:
The Transposition Cipher is another type of encryption method that operates by rearranging the characters or blocks of characters in the plaintext to form the ciphertext. Unlike the Substitution Cipher, which substitutes each letter with another, the Transposition Cipher preserves the original letters but changes their order. One of the well-known examples of a Transposition Cipher is the Railfence Cipher.

Railfence Cipher:
The Railfence Cipher is a basic form of a Transposition Cipher that rearranges the letters of the plaintext by writing them in a zigzag pattern along a set number of "rails." The rails are imaginary horizontal lines on which the plaintext characters are placed.

Encryption Process:
To encrypt a message using the Railfence Cipher, follow these steps:

a) Choose the number of rails (often referred to as the key) for the cipher.
b) Write the plaintext message diagonally along the rails from top to bottom and left to right.
c) Once the last rail is reached, reverse the direction and continue writing diagonally upwards until the first rail is reached again.
d) Read the characters in the zigzag pattern from left to right and from top to bottom to obtain the ciphertext

Decryption Process:
To decrypt a message encrypted with the Railfence Cipher, the recipient needs to know the number of rails (key) used during encryption. The decryption process is the reverse of the encryption process:

a) Write the ciphertext diagonally along the rails, just as it was done during encryption.
b) Read the characters from left to right and from top to bottom to obtain the plaintext.

Code: Python code for implementing Caesar Cipher

```
#A python program to illustrate Caesar Cipher Techniquer
      traverse text
```

```
        for i in range(len(text)):
                char = text[i]

                # Encrypt uppercase characters
                if (char.isupper()):
                        result += chr((ord(char) + s-65) % 26 + 65)

                # Encrypt lowercase characters
                else:
                        result += chr(
```

Code: Java code for implementing Railfence Cipher

```
public class railfence {
    public static void main(String args[])
    {
        String input = "siesnerul";
        String output = "";
        int len = input.length(),flag = 0;

        System.out.println("Input String : " + input);
        for(int i=0;i<len;i+=2) {

            output += input.charAt(i);
        }
        for(int i=1;i<len;i+=2) {

            output += input.charAt(i);
        }

        System.out.println("Ciphered Text : "+output);
    }
}
```