**Experiment 01: Implement and design the product cipher using  
Substitution and Transposition ciphers**

**Experiment 01: (a) Substitution Cipher**

**Learning Objective:** Implement and design the product cipher using Substitution Cipher

**Tools:** PyCharm

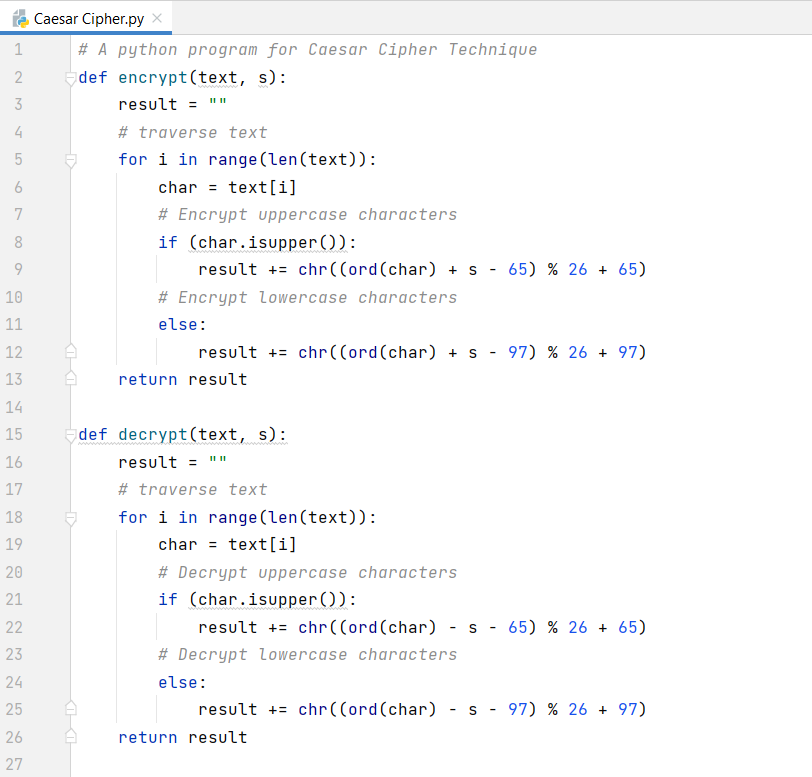
**Theory:**

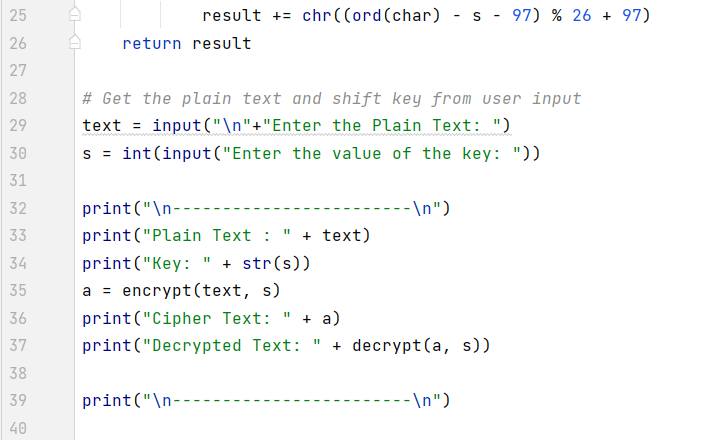
Substitution ciphers are a method of encrypting plaintext by swapping each letter or symbol in the text with a different symbol, based on a specific key. The Caesar cipher is perhaps the simplest and most well-known of these substitution ciphers. It is named after the man who first used it. This cipher is also called a shift cipher or a mono-alphabetic cipher, which differentiates it from other more complex substitution ciphers.

In a Caesar cipher, the plaintext is represented in lowercase letters, while the ciphertext is represented in uppercase letters. Spaces are added to the ciphertext for readability, but they are removed in a real application to make attacking the ciphertext more difficult. Simple substitution of single letters separately can be demonstrated by writing out the alphabet in some order to represent the substitution. This is known as a substitution alphabet. The cipher alphabet can be shifted, reversed, or scrambled in a more complex way to create different types of substitution ciphers.

Mixed alphabets or deranged alphabets can also be used to create substitution ciphers. These are traditionally created by writing out a keyword and removing any repeated letters, then writing all the remaining letters in the alphabet in their usual order. This creates a unique mixed alphabet that can be used as the basis for the cipher. Substitution ciphers have a long history, and although they are not as secure as modern encryption methods, they are still used in some applications today.

**Code:**





**Output:**



**Conclusion:** After performing the experiment I was able to implement Substitution Cipher.

**Experiment 01: (b) Transposition Cipher**

**Learning Objective:** Implement and design the product cipher using Transposition Cipher

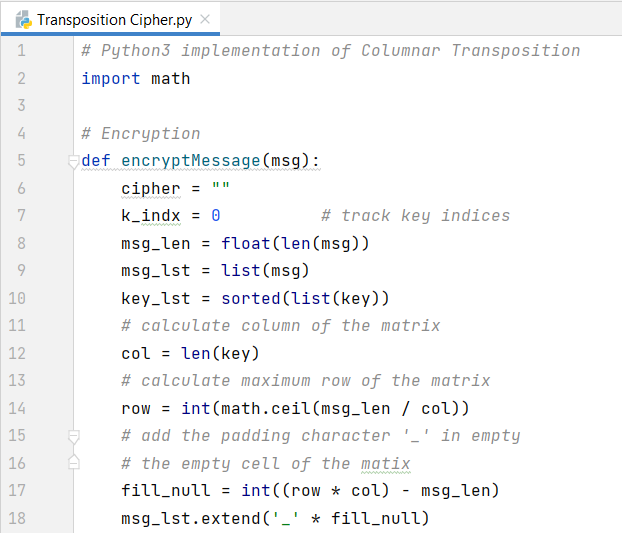
**Tools:** PyCharm

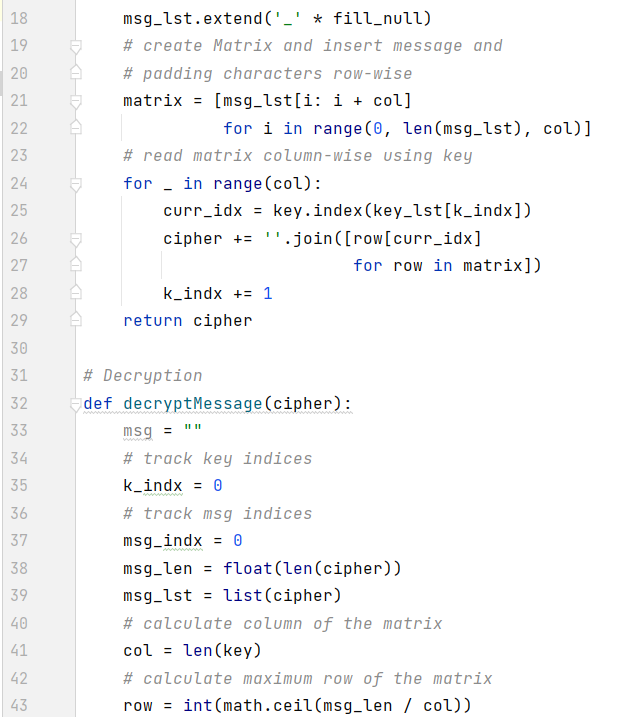
**Theory:**

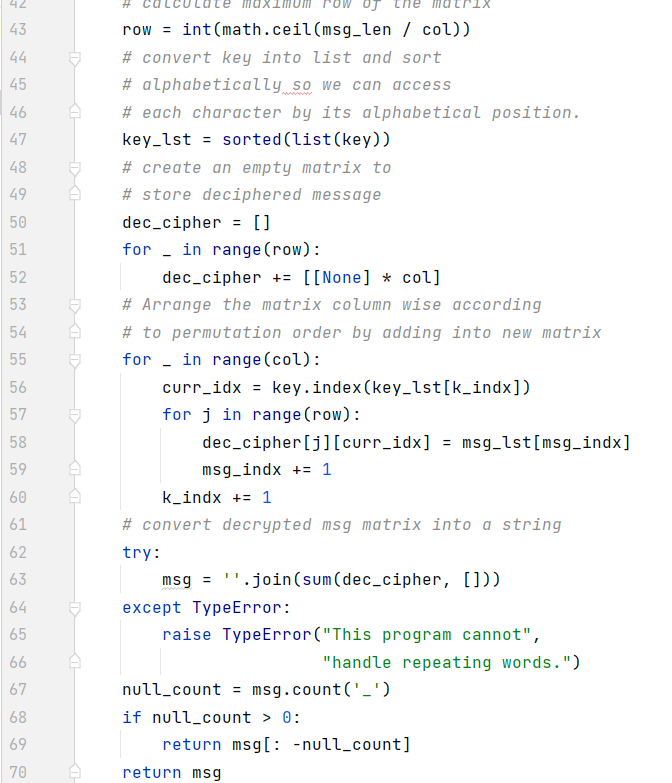
Transposition ciphers are often used in combination with other encryption methods such as substitution ciphers to create a more secure encryption. By adding the additional layer of transposition, the resulting ciphertext becomes much more difficult to decipher without knowledge of both encryption methods. A common method of implementing transposition ciphers is through the use of a rectangular grid, where the plaintext is written out horizontally and then read vertically in a certain order to create the ciphertext. Other methods may involve shuffling the order of words or phrases in the plaintext message.

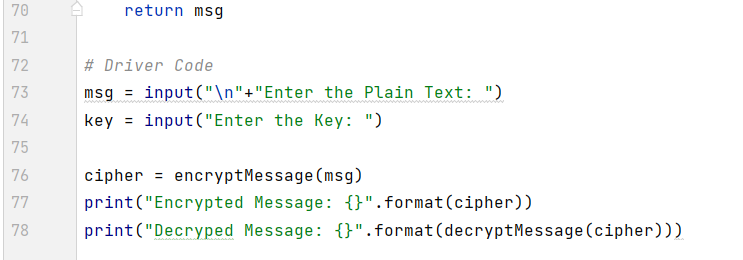
One of the most famous examples of a transposition cipher is the Rail Fence cipher, which involves writing the plaintext diagonally on alternate lines, and then reading the ciphertext vertically. This creates a zig-zag pattern that is difficult to decipher without knowledge of the exact transposition method used. Overall, transposition ciphers offer a flexible and relatively easy method of encryption that can be used in combination with other methods to create a more secure and complex encryption.

**Code:**

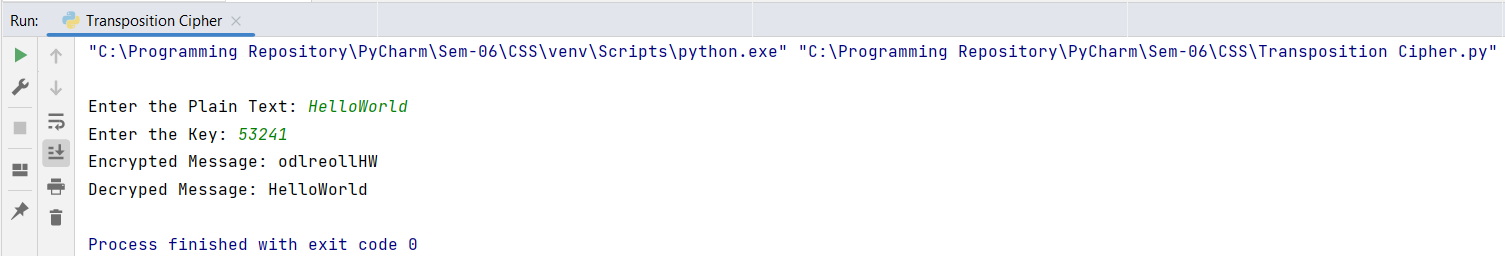








**Output:**

****

**Conclusion:** After performing the experiment I was able to implement Transposition Cipher.

For Faculty Use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning Attitude [20%]** | **Total** |
| **Marks Obtained** |  |  |  |  |