# Experiment 03: Implement and design of Diffie-Hellman Algorithm

**Learning Objective:** Implement and design of Diffie-Hellman Algorithm

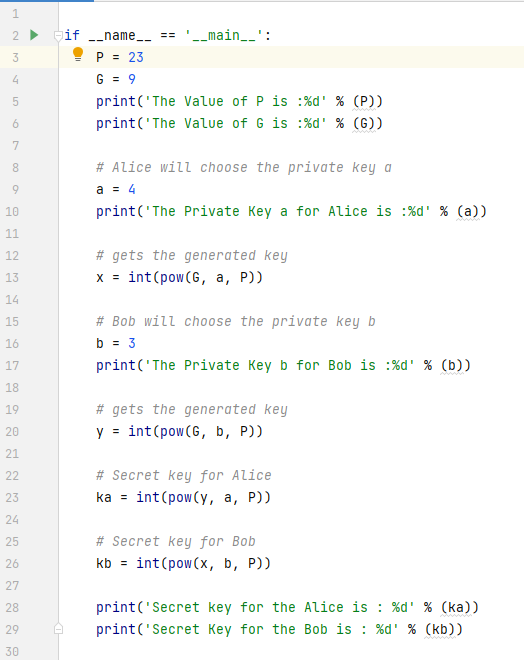
**Tools:** PyCharm

**Theory:**

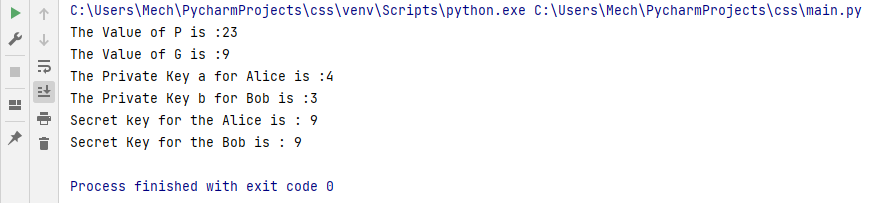
The Diffie–Hellman (DH) Algorithm is a key-exchange protocol that enables two parties communicating over public channel to establish a mutual secret without it being transmitted over the Internet. DH enables the two to use a public key to encrypt and decrypt their conversation or data using symmetric cryptography.

Diffie-Helman is generally explained by two sample parties, Alice and Bob, initiating a dialogue. Each has a piece of information they want to share, while preserving its secrecy. To do that they agree on a public piece of benign information that will be mixed with their privileged information as it travels over an insecure channel. Their secrets are mixed with the public information, or public key, and as the secrets are exchanged the information they want to share is commingled with the common secret. As they decipher the other’s message, they can extract the public information and with knowledge of their own secret, deduce the new information that was carried along. While seemingly uncomplicated in this method’s description, when long number strings are used for private and public keys, decryption by an outside party trying to eavesdrop is mathematically infeasible even with considerable resources.

**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

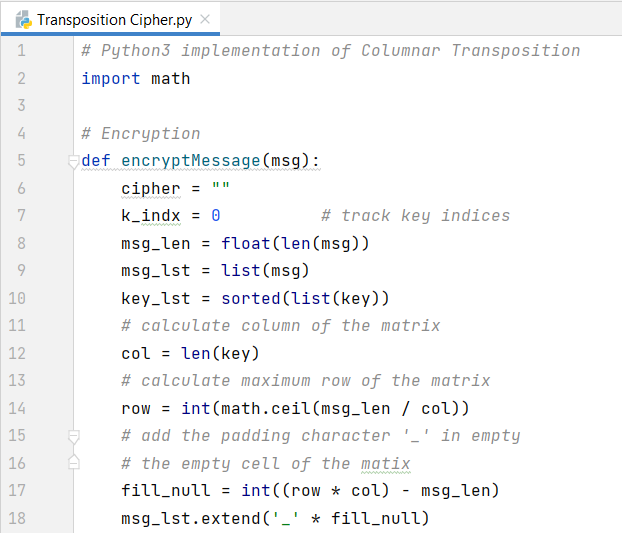
**Tools:** PyCharm

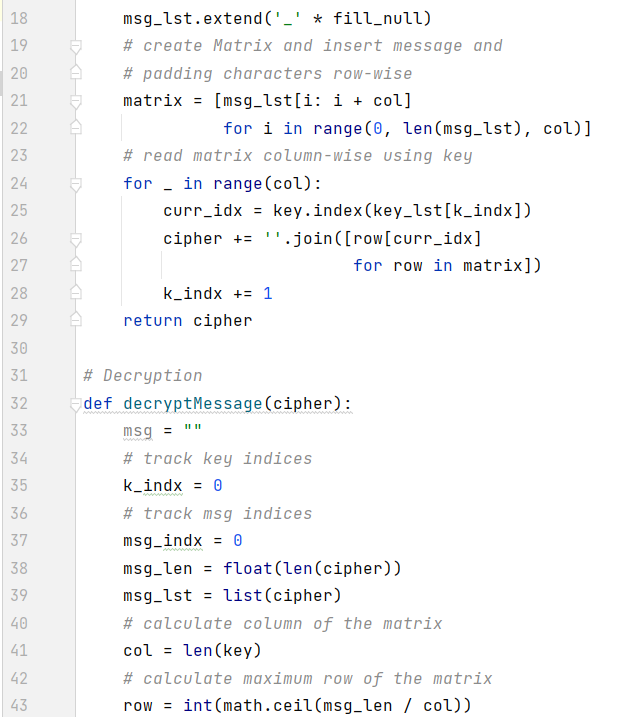
**Theory:**

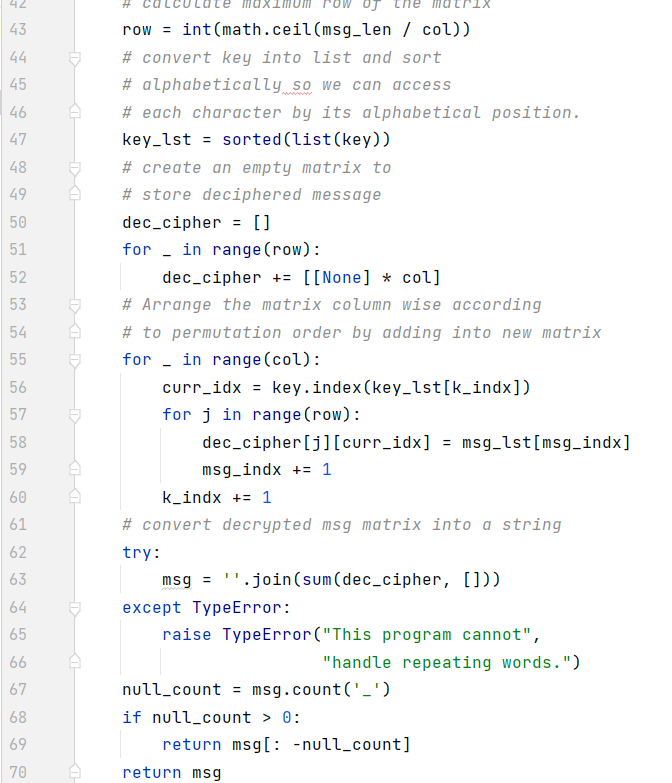
Transposition ciphers are often used in combination with other encryption methods such as substitution ciphers to create a more secure encryption. By adding the additional layer of transposition, the resulting ciphertext becomes much more difficult to decipher without knowledge of both encryption methods. 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.

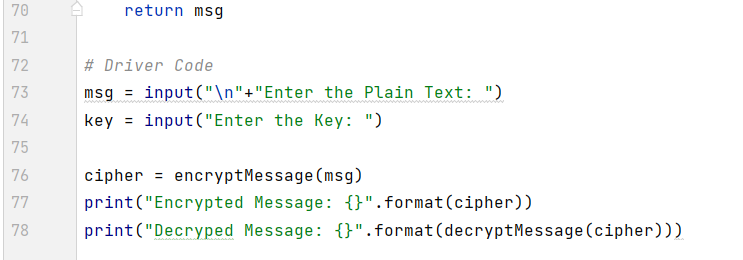
One of the most famous examples of a transposition cipher is the Rail Fence cipher, which involves writing the plaintext diagonally on alternate lines, and then reading the ciphertext vertically. This creates a zig-zag pattern that is difficult to decipher without knowledge of the exact transposition 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:**

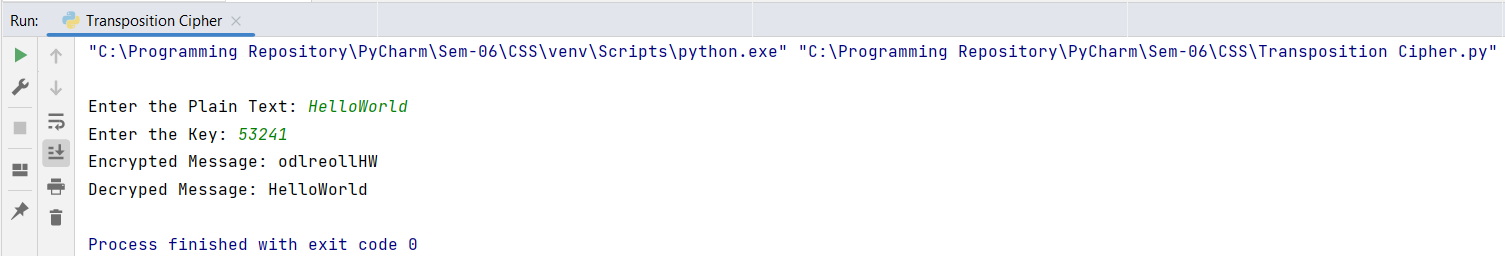








**Output:**

****

**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** |  |  |  |  |