Ques 9. Implement product cipher transposition operation.

Ans:-

```python
# Bharat_Sharma CSC/20/50 Univ_Roll_No: 20059570040
def product_cipher_transposition(plaintext, key):

    key_length = len(key)

    plaintext_length = len(plaintext)

    if plaintext_length % key_length != 0:

        padding_length = key_length - (plaintext_length % key_length)

        plaintext += ' ' * padding_length

        plaintext_length += padding_length


    blocks = [plaintext[i:i+key_length] for i in range(0, plaintext_length, key_length)]


    transposed_blocks = []

    for block in blocks:

        transposed_block = [None] * key_length

        for i, j in enumerate(key):

            transposed_block[j] = block[i]

        transposed_blocks.append(''.join(transposed_block))


    ciphertext = ''.join(transposed_blocks)

    return ciphertext


if __name__ == "__main__":

    plaintext = input("Enter the message : ")

    key = (2, 0, 1)

    ciphertext = product_cipher_transposition(plaintext, key)

    print(ciphertext)
```

_____

OUTPUT:-

jupyter   IS_Ques9 Last Checkpoint: 2 minutes ago (autosaved)   Logout

File   Edit   View   Insert   Cell   Kernel   Help   Trusted   Python 3 (ipykernel) ○

Run   Code

```python
In [2]: # Bharat_Sharma CSC/20/50 Univ_Roll_No: 2005970040
        def product_cipher_transposition(plaintext, key):
            key_length = len(key)
            plaintext_length = len(plaintext)
            if plaintext_length % key_length != 0:
                padding_length = key_length - (plaintext_length % key_length)
                plaintext += ' ' * padding_length
                plaintext_length += padding_length

            blocks = [plaintext[i:i+key_length] for i in range(0, plaintext_length, key_length)]

            transposed_blocks = []
            for block in blocks:
                transposed_block = [None] * key_length
                for i, j in enumerate(key):
                    transposed_block[j] = block[i]
                transposed_blocks.append(''.join(transposed_block))

            ciphertext = ''.join(transposed_blocks)
            return ciphertext

        if __name__ == "__main__":
            plaintext = input("Enter the message : ")
            key = (2, 0, 1)
            ciphertext = product_cipher_transposition(plaintext, key)
            print(ciphertext)
```

```
Enter the message : bharatsharma
habatrhasmar
```

In [ ]: