

```
# Python code to implement
# Vigenere Cipher
```

```
# This function generates the
# key in a cyclic manner until
# it's length isn't equal to
# the length of original text
def generateKey(string, key):
    key = list(key)
    if len(string) == len(key):
        return(key)
    else:
        for i in range(len(string) -
                        len(key)):
            key.append(key[i % len(key)])
    return("".join(key))
```

```
# This function returns the
# encrypted text generated
# with the help of the key
def cipherText(string, key):
    cipher_text = []
    for i in range(len(string)):
        x = (ord(string[i]) +
             ord(key[i])) % 26
        x += ord('A')
        cipher_text.append(chr(x))
    return("".join(cipher_text))
```

```
# This function decrypts the
# encrypted text and returns
# the original text
def originalText(cipher_text, key):
    orig_text = []
    for i in range(len(cipher_text)):
        x = (ord(cipher_text[i]) -
             ord(key[i]) + 26) % 26
        x += ord('A')
        orig_text.append(chr(x))
    return("".join(orig_text))
```

```
# Driver code
if __name__ == "__main__":
```

```
string = "GEEKSFORGEEKS"  
keyword = "AYUSH"  
key = generateKey(string, keyword)  
cipher_text = cipherText(string, key)  
print("Ciphertext :", cipher_text)  
print("Original/Decrypted Text :",  
      originalText(cipher_text, key))
```

This code is contributed

by Pratik Somwanshi