Source Code :

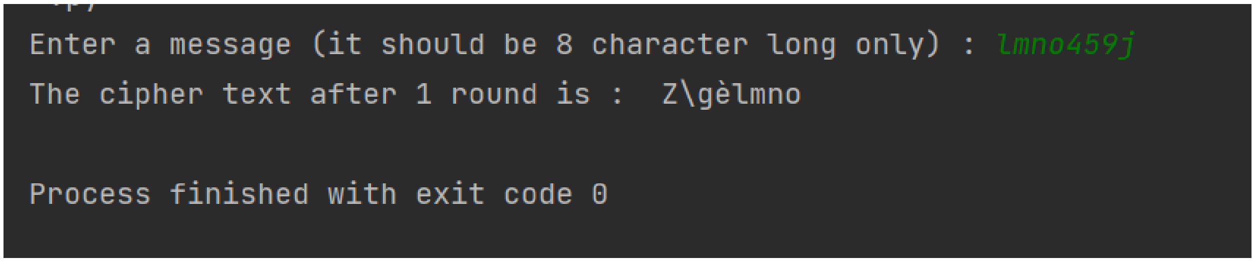
```
import random

def feistelWorking(binary_input: str):
    left = binary_input[:32]
    right = binary_input[32:]
    # print(left + right)
    k = random.randint(1, 42949672)
    k = format(k, '032b')
    xor = []
    for i in range(32):
        xor.append(int(right[i]) ^ int(k[i]))
    new_xor = []
    for i in range(32):
        new_xor.append(int(xor[i]) ^ int(left[i]))
    right = "".join(str(i) for i in new_xor)
    temp = left
    left = right
    right = temp
    total = left + right
    # print(total)
    print("The cipher text after 1 round is : ", end=" ")
    for i in range(0, len(total), 8):
        print(chr(int(total[i:i+8], 2)), end="")
    print()



if __name__ == "__main__":
    message = input("Enter a message (it should be 8 character long only) : ")
    binary_of_message = "".join(format(ord(i), '08b') for i in message)
    feistelWorking(binary_of_message)
    # print(binary_of_message)
```

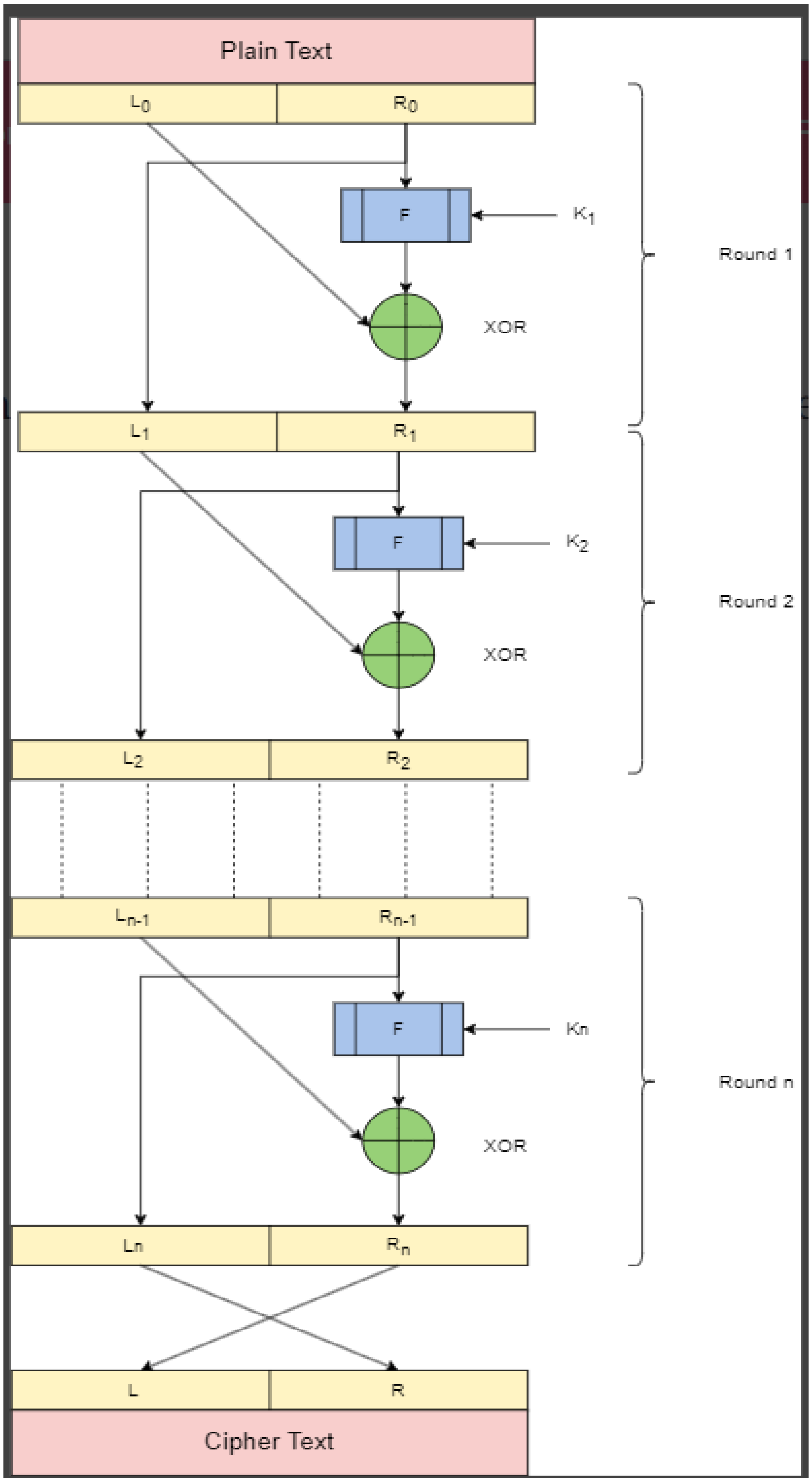
Output :



```
Enter a message (it should be 8 character long only) : lmno459j
The cipher text after 1 round is :  Z\gèlmno


Process finished with exit code 0
```

Feistel cipher structure converts plain text to cipher text using the following steps:


Plain Text
L_0
R_0
F
K_1
XOR
Round 1
L_1
R_1
F
K_2
XOR
Round 2
L_2
R_2
L_{n-1}
R_{n-1}
F
K_n
XOR
Round n
L_n
R_n
L
R
Cipher Text