

Basic Reversing

```
def encode(input_data, key):  
    return ''.join(f"{ord(input_data[i]) ^ ord(key[i % len(key)])}"  
  
def main():  
    flag = "JFS{***terminated***}"  
    key = input('Input key: ')  
    encoded_data = encode(flag, key)  
    if len(key) == 5:  
        if encoded_data == "7e14144b667b0d0271666d0d157563710014797b7373661148":  
            print("Correct!!")  
        else :  
            print("Wrong..")  
    else:  
        print("Invalid Key length")  
  
main()
```

주어진 python 코드를 보면 KEY를 입력 받고 입력 받은 key를 가지고 xor 연산을 하여 encoding을 수행하는 것을 알 수 있다.

"7e14144b667b0d0271666d0d157563710014797b7373661148" 이는 encoding된 hex 값으로 알 수 있고

encode() 함수로 보아 항상 2자리씩 출력이 됨을 알 수 있다. 즉, 이 hex의 길이 / 2는 원래 flag의 길이 임을 알 수 있다.

```
len(FLAG) == 25
```

이때 Key를 알아내면 flag를 복구할 수 있다.

Key는 길이가 5이면서

flag format JFS{...}와 xor 연산을 하여 encoded_data를 생성한다.

key에 대한 정보로는

```
ord(key[0]) ^ ord("J") = 0x7e
ord(key[1]) ^ ord("F") = 0x14
ord(key[2]) ^ ord("S") = 0x14
ord(key[3]) ^ ord("{") = 0x4b
ord(key[4]) ^ ord("}") = 0x48
```

을 얻을 수 있다.

```
# solve.py
key = ""

key += chr(ord("J") ^ 0x7e)
key += chr(ord("F") ^ 0x14)
key += chr(ord("S") ^ 0x14)
key += chr(ord("{") ^ 0x4b)
key += chr(ord("}") ^ 0x48)

print('KEY:', key)
```

```
argos@gameserver:~/semon/jfs/pycode$ python3 solve.py
KEY: 4RG05
```

KEY : 4RG05

이 찾은 KEY를 가지고 decoding을 하면

```
target = "7e14144b667b0d0271666d0d157563710014797b7373661148"
flag = ""

for i in range(len(target)//2):
    flag += chr(ord(key[i % len(key)]) ^ int(target[i*2:i*2+2], 16))

print('FLAG:', flag)
```

위와 같이 코드를 만들 수 있고

결과는 다음과 같다.

```
argos@gameserver:~/semon/jfs/pycode$ python3 solve.py
KEY: 4RG05
FLAG: JFS{SO_EASY_REVERSING!!!}
```

전체 풀이 코드

```
# solve.py
key = ""

key += chr(ord("J") ^ 0x7e)
key += chr(ord("F") ^ 0x14)
key += chr(ord("S") ^ 0x14)
key += chr(ord("{") ^ 0x4b)
key += chr(ord("}") ^ 0x48)

print('KEY:', key)

target = "7e14144b667b0d0271666d0d157563710014797b7373661148"
flag = ""

for i in range(len(target)//2):
    flag += chr(ord(key[i % len(key)]) ^ int(target[i*2:i*2+2]))

print('FLAG:', flag)
```

```
FLAG : JFS{SO_EASY_REVERSING!!!}
```