

Since xor() is just a basic xor, solving this revolves in the property that a XOR a = 0 and 0 XOR a = a

So if we xor the string we receive we get a flag XOR encoded_data.

So to get the flag we xor the 2 output strings from the service with our encoded data:

```
output =
b'\xab\xdb\xc7u]\x02\xb7h\xf8\x93\x04\x19\xf2T\xe2\x16\xf2\xa2D\x1f4\xe8\x91g\xda
\xbd\xa2B\x1c\xb7]"\\xb1y\x1b\xee\x14\x1e\x9e\xfc\xe5Wh%\xad~\xf1\xc6\x11\x0b\xc
cA\xebD\xcc\xb4\x15\x0c9\xed\xafq\x8a\xa8\xab\x13\x08\xe1cpS\xe1\x7f\x17\xbbJ\x02
'

flag_length = 39
encrypted_flag = output[:flag_length]
print(encrypted_flag)
encrypted_data = output[flag_length:]
print(encrypted_data)
user_input =
'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa'.encode()
retrieved_flag = bytes([encrypted_flag[i] ^ encrypted_data[i] for i in
range(len(encrypted_flag))])
final_flag = bytes([retrieved_flag[i] ^ user_input[i % len(user_input)] for i in
range(len(retrieved_flag))])
print("The original flag is:", final_flag.decode())
```

I had to guess the flag_length, so I tried various values until it decoded it.

Made with love by: AndreiCat