

Our dog doesn't byte

```
Disassembling function: encode1
6          0 LOAD_GLOBAL          0 (list)
           2 LOAD_FAST             0 (data)
           4 CALL_FUNCTION         1
           6 STORE_FAST           0 (data)

7          8 LOAD_GLOBAL          1 (range)
          10 LOAD_GLOBAL          2 (len)
          12 LOAD_FAST             0 (data)
          14 CALL_FUNCTION         1
          16 CALL_FUNCTION         1
          18 GET_ITER
      >> 20 FOR_ITER              24 (to 70)
          22 STORE_FAST           1 (i)

8          24 LOAD_GLOBAL          3 (chr)
          26 LOAD_GLOBAL          4 (ord)
          28 LOAD_FAST             0 (data)
          30 LOAD_FAST             1 (i)
          32 BINARY_SUBSCR
          34 CALL_FUNCTION         1
          36 LOAD_CONST            1 (4)
          38 BINARY_LSHIFT
          40 LOAD_GLOBAL          4 (ord)
          42 LOAD_FAST             0 (data)
          44 LOAD_FAST             1 (i)
          46 BINARY_SUBSCR
          48 CALL_FUNCTION         1
          50 LOAD_CONST            1 (4)
          52 BINARY_RSHIFT
          54 BINARY_OR
          56 LOAD_CONST            2 (255)
          58 BINARY_AND
          60 CALL_FUNCTION         1
```

		62	LOAD_FAST	0	(data)
		64	LOAD_FAST	1	(i)
		66	STORE_SUBSCR		
		68	JUMP_ABSOLUTE	10	(to 20)
10	>>	70	LOAD_FAST	0	(data)
		72	RETURN_VALUE		

Disassembling function: encode2

13		0	LOAD_CONST	1	('woof')
		2	STORE_FAST	1	(key)
14		4	LOAD_CONST	2	('')
		6	STORE_FAST	2	(result)
16		8	LOAD_GLOBAL	0	(range)
		10	LOAD_GLOBAL	1	(len)
		12	LOAD_FAST	0	(data)
		14	CALL_FUNCTION	1	
		16	CALL_FUNCTION	1	
		18	GET_ITER		
	>>	20	FOR_ITER	22	(to 66)
		22	STORE_FAST	3	(i)
17		24	LOAD_FAST	2	(result)
		26	LOAD_GLOBAL	2	(chr)
		28	LOAD_GLOBAL	3	(ord)
		30	LOAD_FAST	0	(data)
		32	LOAD_FAST	3	(i)
		34	BINARY_SUBSCR		
		36	CALL_FUNCTION	1	
		38	LOAD_GLOBAL	3	(ord)
		40	LOAD_FAST	1	(key)
		42	LOAD_FAST	3	(i)
		44	LOAD_GLOBAL	1	(len)
		46	LOAD_FAST	1	(key)
		48	CALL_FUNCTION	1	
		50	BINARY_MODULO		

```

52 BINARY_SUBSCR
54 CALL_FUNCTION          1
56 BINARY_XOR
58 CALL_FUNCTION          1
60 INPLACE_ADD
62 STORE_FAST             2 (result)
64 JUMP_ABSOLUTE          10 (to 20)

18      >> 66 LOAD_CONST             2 (')
68 LOAD_METHOD            4 (join)
70 LOAD_CONST             3 (<code object <gen
72 LOAD_CONST             4 ('encode2.<locals>
74 MAKE_FUNCTION          0
76 LOAD_FAST              2 (result)
78 GET_ITER
80 CALL_FUNCTION           1
82 CALL_METHOD            1
84 STORE_FAST             2 (result)

19      86 LOAD_FAST          2 (result)
88 RETURN_VALUE

```

Disassembly of <code object <genexpr> at 0x7f72c0ae7b50, file

```

0 GEN_START          0

18      2 LOAD_FAST          0 (.0)
      >> 4 FOR_ITER            9 (to 24)
        6 STORE_FAST        1 (i)
        8 LOAD_GLOBAL       0 (ord)
       10 LOAD_FAST          1 (i)
       12 CALL_FUNCTION       1
       14 LOAD_CONST         0 ('02x')
       16 FORMAT_VALUE        4 (with format)
       18 YIELD_VALUE
       20 POP_TOP
       22 JUMP_ABSOLUTE       2 (to 4)
      >> 24 LOAD_CONST         1 (None)
       26 RETURN_VALUE

```

Disassembling function: main

22	0	LOAD_GLOBAL	0 (input)	
	2	LOAD_CONST	1 ('Input: ')	
	4	CALL_FUNCTION	1	
	6	STORE_FAST	0 (user_input)	
23	8	LOAD_GLOBAL	1 (encode1)	
	10	LOAD_FAST	0 (user_input)	
	12	CALL_FUNCTION	1	
	14	STORE_FAST	1 (encoded_data)	
24	16	LOAD_GLOBAL	2 (encode2)	
	18	LOAD_FAST	1 (encoded_data)	
	20	CALL_FUNCTION	1	
	22	STORE_FAST	2 (result)	
26	24	LOAD_FAST	2 (result)	
	26	LOAD_CONST	2 ('d30b5ad1936c2893	
	28	COMPARE_OP	2 (==)	
	30	POP_JUMP_IF_FALSE	22 (to 44)	
27	32	LOAD_GLOBAL	3 (print)	
	34	LOAD_CONST	3 ('Correct!')	
	36	CALL_FUNCTION	1	
	38	POP_TOP		
	40	LOAD_CONST	0 (None)	
	42	RETURN_VALUE		
29	>>	44	LOAD_GLOBAL	3 (print)
		46	LOAD_CONST	4 ('Wrong..')
		48	CALL_FUNCTION	1
		50	POP_TOP	
		52	LOAD_CONST	0 (None)
		54	RETURN_VALUE	

주어진 bytecode를 크게 4부분으로 나누면

```
Disassembling function: encode1
6          0 LOAD_GLOBAL          0 (list)
           2 LOAD_FAST              0 (data)
           4 CALL_FUNCTION        1
           6 STORE_FAST          0 (data)

7          8 LOAD_GLOBAL          1 (range)
          10 LOAD_GLOBAL          2 (len)
          12 LOAD_FAST              0 (data)
          14 CALL_FUNCTION        1
          16 CALL_FUNCTION        1
          18 GET_ITER
      >> 20 FOR_ITER              24 (to 70)
          22 STORE_FAST          1 (i)

8          24 LOAD_GLOBAL          3 (chr)
          26 LOAD_GLOBAL          4 (ord)
          28 LOAD_FAST              0 (data)
          30 LOAD_FAST              1 (i)
          32 BINARY_SUBSCR
          34 CALL_FUNCTION        1
          36 LOAD_CONST            1 (4)
          38 BINARY_LSHIFT
          40 LOAD_GLOBAL          4 (ord)
          42 LOAD_FAST              0 (data)
          44 LOAD_FAST              1 (i)
          46 BINARY_SUBSCR
          48 CALL_FUNCTION        1
          50 LOAD_CONST            1 (4)
          52 BINARY_RSHIFT
          54 BINARY_OR
          56 LOAD_CONST            2 (255)
          58 BINARY_AND
          60 CALL_FUNCTION        1
          62 LOAD_FAST              0 (data)
```

		64 LOAD_FAST	1 (i)
		66 STORE_SUBSCR	
		68 JUMP_ABSOLUTE	10 (to 20)
10	>>	70 LOAD_FAST	0 (data)
		72 RETURN_VALUE	

Disassembling function: encode2

13		0 LOAD_CONST	1 ('woof')
		2 STORE_FAST	1 (key)
14		4 LOAD_CONST	2 ('')
		6 STORE_FAST	2 (result)
16		8 LOAD_GLOBAL	0 (range)
		10 LOAD_GLOBAL	1 (len)
		12 LOAD_FAST	0 (data)
		14 CALL_FUNCTION	1
		16 CALL_FUNCTION	1
		18 GET_ITER	
	>>	20 FOR_ITER	22 (to 66)
		22 STORE_FAST	3 (i)
17		24 LOAD_FAST	2 (result)
		26 LOAD_GLOBAL	2 (chr)
		28 LOAD_GLOBAL	3 (ord)
		30 LOAD_FAST	0 (data)
		32 LOAD_FAST	3 (i)
		34 BINARY_SUBSCR	
		36 CALL_FUNCTION	1
		38 LOAD_GLOBAL	3 (ord)
		40 LOAD_FAST	1 (key)
		42 LOAD_FAST	3 (i)
		44 LOAD_GLOBAL	1 (len)
		46 LOAD_FAST	1 (key)
		48 CALL_FUNCTION	1

```

50 BINARY_MODULO
52 BINARY_SUBSCR
54 CALL_FUNCTION          1
56 BINARY_XOR
58 CALL_FUNCTION          1
60 INPLACE_ADD
62 STORE_FAST             2 (result)
64 JUMP_ABSOLUTE         10 (to 20)

18      >> 66 LOAD_CONST          2 (')
          68 LOAD_METHOD          4 (join)
          70 LOAD_CONST          3 (<code object <gen
          72 LOAD_CONST          4 ('encode2.<locals>
          74 MAKE_FUNCTION        0
          76 LOAD_FAST           2 (result)
          78 GET_ITER
          80 CALL_FUNCTION        1
          82 CALL_METHOD         1
          84 STORE_FAST          2 (result)

19      86 LOAD_FAST           2 (result)
          88 RETURN_VALUE

```

Disassembly of <code object <genexpr> at 0x7f934cee4920, file

```

      0 GEN_START          0

18      2 LOAD_FAST           0 (.0)
      >> 4 FOR_ITER            9 (to 24)
          6 STORE_FAST          1 (i)
          8 LOAD_GLOBAL         0 (ord)
         10 LOAD_FAST           1 (i)
         12 CALL_FUNCTION        1
         14 LOAD_CONST          0 ('02x')
         16 FORMAT_VALUE         4 (with format)
         18 YIELD_VALUE
         20 POP_TOP

```

```

                22 JUMP_ABSOLUTE          2 (to 4)
>>            24 LOAD_CONST              1 (None)
                26 RETURN_VALUE

```

Disassembling function: main

```

22             0 LOAD_GLOBAL              0 (input)
                2 LOAD_CONST              1 ('Input: ')
                4 CALL_FUNCTION          1
                6 STORE_FAST            0 (user_input)

23             8 LOAD_GLOBAL              1 (encode1)
            10 LOAD_FAST                0 (user_input)
            12 CALL_FUNCTION          1
            14 STORE_FAST            1 (encoded_data)

24            16 LOAD_GLOBAL              2 (encode2)
            18 LOAD_FAST                1 (encoded_data)
            20 CALL_FUNCTION          1
            22 STORE_FAST            2 (result)

26            24 LOAD_FAST                2 (result)
            26 LOAD_CONST              2 ('d30b5ad1936c2893')
            28 COMPARE_OP              2 (==)
            30 POP_JUMP_IF_FALSE        22 (to 44)

27            32 LOAD_GLOBAL              3 (print)
            34 LOAD_CONST              3 ('Correct!')
            36 CALL_FUNCTION          1
            38 POP_TOP
            40 LOAD_CONST              0 (None)
            42 RETURN_VALUE

29    >>      44 LOAD_GLOBAL              3 (print)
                46 LOAD_CONST              4 ('Wrong..')
                48 CALL_FUNCTION          1
                50 POP_TOP

```


52	LOAD_CONST	0 (None)
54	RETURN_VALUE	

다음과 같이 나뉘지고 각각 함수의 형태를 띄게 된다.

main은 input을 기준으로 encode1과 encode2를 호출하고 그 결과 값을

`d30b5ad1936c289353f8283082cb3851049a4b7050d93c7465b8` 와 비교하여 같으면 Correct 틀리면 Wrong을 출력한다.

encode1은 bit swap을 나타내고 encode2는 key와 xor연산을 수행하며 중간에 다른 함수는 이 결과를 hex로 출력하는 역할을 한다. 즉, 이를 역연산하면

```
target = "d30b5ad1936c289353f8283082cb3851049a4b7050d93c7465b8"
key = "woof"
result = ""
for i in range(len(target)//2):
    result += chr(int(target[i*2:i*2+2], 16) ^ ord(key[i % len(key)]))

result = list(result)

for i in range(len(result)):
    result[i] = chr((ord(result[i]) << 4 | ord(result[i]) >> 4))

print(''.join(i for i in result))
```

위와 같은 코드가 나오고

이를 실행하면 flag를 획득할 수 있다.

```
argos@gameserver:~/semon/jfs/pycode$ python3 solve_byte.py
JFS{N0t_Byte_Jus7_Bark5!!}
```

FLAG : JFS{N0t_Byte_Jus7_Bark5!!}