

Is that Secure ?

Bob is sending a message to Alice, he encrypts his message using the following algorithm :
Let S be a plaintext message of length n , R be the encrypted message, for every character S_i of S , he first computes the **xor** product of the ascii codes of all the other characters of S_i , that is, he computes $v_i = S_0 \text{ xor } S_1 \text{ xor } \dots \text{ xor } S_{i-1} \text{ xor } S_{i+1} \text{ xor } \dots \text{ xor } S_{n-1}$, then he inserts the hex value of v_i (on two digits) on the i_{th} position in the string R .

For example, if $S = \text{"Crypto"}$, the **result** of the encryption will be $R = \text{"60515a53574c"}$.

He also generates a key that is needed to decrypt the message, but its algorithm is **not** known to us, then he sends the encrypted message and the key to Alice.

*The Plain text message can **only** contain lowercase, uppercase alphabet, digits, spaces, commas, question marks, and exclamation marks.*

Given this **list** of messages, the answer will be a **list of comma-separated** plain text messages.

```
6956511e49514b525a1e4c5b5f5a1e4a56574d01
305915160f1c5900160c593815101a1c58
2a3c353c3a2d792c2a3c2b3738343c757929382a2a2e362b3d793f2b3634792c2a3c2b2a
79184b6f5968185b794b7d18557d4b6b597f5d14187b59761841774d186a5d795c18714c07
203c317424352727233b2630743d2774656667606162636c
71667366776a776a6c6d
7a5b185e414740145a5b
```

Challenge Points : 200 pts

Happy coding.