ECOO '12 R2 P1 - Prime Time

Time limit: 2.0s **Memory limit:** 64M

Alice is sending Bob coded messages using an encoding scheme of her own design. When she wants to send a message, she starts by turning the characters of the message into integers. This is done by assigning 0 to the space character, then 1 to `A`, 2 to `B`, and so on through the alphabet. She also assigns 27 to the period character, 28 to the comma, 29 to the exclamation mark, and 30 to the question mark.

After Alice converts characters to numbers, she puts each pair of numbers together to make a new 4-digit number (if she needs to, she pads the message with an extra space at the end). Then she selects a six-digit prime number less than $500\,000$ as the "key" value. She multiplies each of the 4-digit numbers in the message by the key and transmits the resulting numbers, starting with an un-encoded integer that indicates how many coded numbers are coming.

Example:

`HEY BOB!` → `HE` + `Y` + `BO` + `B!` → `0805 2500 0215 0229`

Key: $395\,611$
Encoded Message: `4 318466855 989027500 85056365 90594919`

The genius part of this scheme is that when Alice sends Bob a message, Bob doesn't need to know the key ahead of time. So she is free to choose any key she wants and she can change the key for each message. Unfortunately, the "genius" part also makes her coded messages very easy for a third party to crack.

The input contains five coded messages from Alice to Bob (minimum length 4 characters per message). Your job is to decode each message and display the results, one message per line.

Sample Input

```
4 318466855 989027500 85056365 90594919
6 904474779 361632680 1100621200 907226332 47169480 1179237000
4 307114756 570239600 307725727 549873900
8 149471983 450198915 810358047 802334700 149471983 450198915 810358047 802334700
7 105593497 195549029 74466500 14893300 32467394 74615433 168145357
```

Sample Output

```
HEY BOB!
WAIT, WHAT?
OH, OK.
ECOO... ECOO...
GIMME A BREAK!
```