

B - Hide That Number

Input: standard input

Output: standard output

According to Wikipedia, *Cryptography* is “the practice and study of hiding information” and this is exactly what Alex is looking for. Ever since he was a kid, Alex was paranoid over someone laying their hands on his phone book. He decided that he must write the numbers in some secret way that only he can decipher.

At first he tried quite complex algorithms, but that slowed him down when he needed to dial a number fast. He finally came up with the following algorithm: Rather than writing down the number itself, Alex would shift the number one place to the left (as if multiplying it by 10,) then adding the shifted number to the original. For example, if the phone number was 123, Alex would add 1230 to it, resulting in 1353. To make what he writes look like a regular phone number, Alex truncates the result (from the left,) so that it has as many digits as the original phone number. In this example, Alex writes 353 instead of 123 in his phone book.

Alex needs a program to print the original phone number given what is written in his phone book. Alex, who by the way is a good friend of Johnny, isn’t that good in arithmetic. It is quite possible that the numbers are messed up. The program should print “IMPOSSIBLE” (without the quotes) if the original number cannot be computed.

Input

Your program will be tested on one or more test cases. Each test case is specified on a separate line and is made of a single positive number having less than 1,000,000 digits.

The last line of the input file is made of a single zero.

Output

For each test case, output the result on a single line using the following format:

`k. _result`

Where k is the test case number (starting at 1) and `_` is a single space.

Sample Input

```
353
9988
123456
0
```

Sample Output

```
1. 123
2. IMPOSSIBLE
3. 738496
```