

# Pokechat

Problem ID: pokechat

The year is 2011, and you and your friend each got a copy of the newly released Pokemon Diamond and Pearl games. You enjoy playing the game together, but you can't always be together and talk while you're playing because your mom makes you come home after dark. This would be fine, except that the in-game chat functionality only allows for predetermined messages and you want to talk freely with your friend while you play. In order to be able to communicate any message, you and your friend come up with a simple code to convey characters to each other by displaying Pokemon using the game's trading feature.

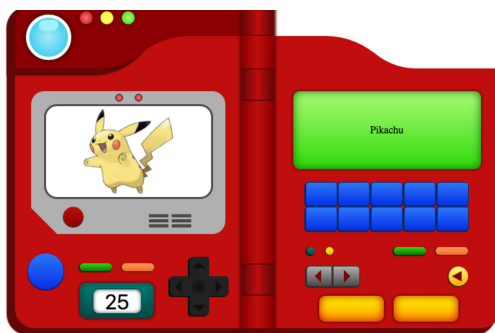


Figure 1: A real working Pokedex!!!

The code takes advantage of an encoding string  $S$ , which contains a series of unique characters, which you and your friend decided on in advance and both have a copy of. By displaying a Pokemon of id  $n$ , your friend (or you) is signaling that they intend to communicate the  $n$ -th character of the encoding string that you are both using as a code. By displaying specific Pokemon in a certain order, with one Pokemon per character in the message you want to send, you and your friend can then take turns communicating full messages of your own choices. For example, with an encoding string of *abr*, the id number ordering of 001, 002, 003, and 001 represent *abra*. Given the encoding string, and a string of Pokemon ids that your friend has displayed in the order given by the string, determine the corresponding message that your friend has sent to you.

## Input

A string  $S$  of unique characters representing the encoding string, where each character is a printable ASCII character (note that the whitespace character could be in  $S$ ). It is guaranteed that the length of  $S$  will be less than 100.

A non-empty string of 3-digit Pokemon id numbers (zeros will be added in front of 1 and 2 digit id numbers) of length  $l$ , where  $0 < l < 1000$  and  $l$  is divisible by 3. It is guaranteed that each Pokemon id number will be less than or equal to the length of  $S$ .

## Output

A non-empty message consisting only of characters found in the encoding string.

### Sample Input 1

```
PpIiKkAaCcHhUu  
001004006008010012014
```

### Sample Output 1

```
Pikachu
```

### Sample Input 2

```
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .,?!0123456789  
016009011001053016009011001
```

### Sample Output 2

```
pika pika
```