ACM Qualifier 2016: A New Alphabet

A New Alphabet has been developed for Internet communications. While the glyphs of the new alphabet don't necessarily improve communications in any meaningful way, they certainly make us feel cooler.

You are tasked with creating a translation program to speed up the switch to our more elite New Alphabet by automatically translating ASCII plaintext symbols to our new symbol set.

The new alphabet is a one-to-many translation (one character of the English alphabet translates to anywhere between 1 and 6 other characters), with each character translation as follows:

Original	New	English Description	Original	New	English Description
a	@	at symbol	n	[]\[]	brackets, backslash, brackets
b	8	digit eight	o	0	digit zero
c	(	open parenthesis	p	|D	bar, capital D
d	|)	bar, close parenthesis	q	(,)	parenthesis, comma, parenthesis
e	3	digit three	r	|Z	bar, capital Z
f	#	number sign (hash)	s	$	dollar sign
g	6	digit six	t	']['	quote, brackets, quote
h	[-]	bracket, hyphen, bracket	u	|_|	bar, underscore, bar
i	|	bar	v	\/	backslash, forward slash
j	_|	underscore, bar	w	\/\/	four slashes
k	|<	bar, less than	x	}{	curly braces
l	1	digit one	y	`/	backtick, forward slash
m	[]\/[]	brackets, slashes, brackets	z	2	digit two

For instance, translating the string "Hello World!" would result in:

```
[-]3110 \/\/0|Z1|)!
```

Note that uppercase and lowercase letters are both converted, and any other characters remain the same (the exclamation point and space in this example).

Input Format

Input contains one line of text, terminated by a newline. The text may contain any characters in the ASCII range 32–126 (space through tilde), as well as 9 (tab). Only characters listed in the above table (A–Z, a–z) should be translated; any non-alphabet characters should be printed (and not modified). Input has at most 10 000 characters.

Output Format

Output the input text with each letter (lowercase and uppercase) translated into its New Alphabet counterpart.

Sample Input

```
All your base are belong to us.
```

Sample Output

```
@11 '/0|_||Z 8@$3 @|Z3 8310[]\[]6 ']['0 |_|$.
```