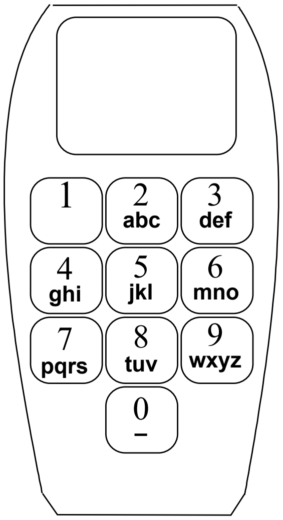
# CCC '06 J3 - Cell-Phone Messaging

**Time limit:** 2.0s **Memory limit:** 256M

### Canadian Computing Competition: 2006 Stage 1, Junior #3



Joe Coder has just received a cell phone for his birthday. At first he was not so excited about it, since he does not like to talk that much, nor listen for that matter, and he hates being interrupted by phone calls while coding or playing his favourite computer game. But, Joe learned that he can talk to his friends and also send e-mails. That made the phone cool.

In order to fit the 26 letters of the alphabet onto the keys of a cell phone, several letters are assigned to each key, as shown on the diagram. To write a text message, we have to choose a letter from a set assigned to a key. The first letter on a key is chosen by pressing the key once, the second letter by pressing the key twice, third letter by pressing the key three times, and so on.

For example, to write `a` we press the key 2 once and we are done; to write `dada` we press 3232—four key presses; and to write `bob` we press 2266622.

An obvious issue is how to write two consecutive letters on the same key, for example in `abba` or `cell`. The problem is solved by introducing a time-out feature: a letter currently displayed is chosen when another key is pressed, but also after a pause, i.e., a time out. Hence for example, to write `abba` we press 2-pause-22-pause-22-pause-2; to write `cell` we press 22233555-pause-555; or to write `www` we press 9-pause-9-pause-9.

This kind of typing takes some time, and Joe is working on a program to calculate how much time is needed to type certain words. His assumption is that he spends one second per press, and whenever he makes a pause he loses an additional two seconds. You are to help him to calculate the minimal time needed to type a message, under the above assumptions.

## Input

Each line of input contains a word consisting only of lowercase letters. Words have at most 20 characters. Input will be given from the keyboard, and the program should stop reading input when the word `halt` has been entered.

## Output

For each input word (excluding the word `halt`), print (on the screen) the minimal number of seconds needed to type in the word, with one number of output per line.

## Sample Input

```
a
dada
bob
abba
cell
www
halt
```

## Sample Output

```
1
4
7
12
13
7
```