


iBoard

Problem ID: iboard
CPU Time limit: 1 second
Memory limit: 1024 MB
Difficulty: 2.6

Author(s): David Sturgill
Source: Baylor Competitive Learning course
License: 



After years of success with a single-button mouse, a well known computer company has decided to offer a similarly simplified interface for the keyboard. The iBoard has only two keys. The user types by entering the ASCII code for each letter. To help people type faster, the iBoard uses key press and key release transitions to enter characters. Pressing or releasing the right key enters zero for the next bit. Pressing or releasing the left key enters a one. The binary code for each 7-bit ASCII character is entered starting with the least significant bit and working up to the most significant bit. For example, to type a capital Z, the user simply has to press the right key, press the left key, release the right key, release the left key, press the left key, press the right key and then release the left key.

Although its creators would like you to believe that the iBoard is far superior to any other keyboard, it does have one disadvantage. The operator may not be free to get up and go to the bathroom as soon as he or she needs to. After typing a string of characters, the user may be stuck with one or more fingers pressing iBoard keys. Releasing these keys would begin typing the next character. Your job is to write a program that determines when it's safe to get up and leave the keyboard. Your program should read input strings and report if it's safe to get up and leave the iBoard after typing each string.

Input

Input contains up to 100 strings, one per line, terminated at end of file. Each string has between 1 and 100 ASCII characters. End-of-line characters should not be considered part of each string.

Output

For each input string, print out a line reporting either `trapped` or `free`. The user is considered trapped if one or more iBoard keys is still being pressed after the string is typed. Otherwise, the user is free. You should assume that no iBoard keys are being pressed when the user starts typing the string.

Sample Input 1

```
Keep up the good work.  
iBoard Rules!!  
qwerty
```

Sample Output 1

```
trapped  
free  
trapped
```