

## Problem D - Snap

*Snap* is a 2-player card game. The deck of cards contains several of each type of card. Initially each player has one half of the deck, in some random sequence, face down in a pile, and plays them in sequence from top to bottom, placing each card face-up in another pile. When the face-down pile is exhausted, the face-up pile is turned over to become the face-down pile and play continues.

The two players play their cards concurrently and synchronously. That is, each places a card face up at exactly the same time. If two cards turned up at the same time are the same type, each player calls "Snap!" and the player who calls first takes the other's face-up pile and places it on top of his or her own.

Play proceeds until one player has all the cards. This player is the winner.

Your job is to simulate a game of snap to determine whether it will end within 1000 turns and, if so, to determine the winner.

Each type of card is denoted by a single letter or digit. The first line of input Jane's initial pile of cards, from top to bottom. The second line of input is John's initial pile. Jane and John start with the same number of cards; not more than 50 each.

During play, whenever it comes time to call "Snap!" the builtin function "random" is used to determine who calls first: if the expression

`random()/141%2` {in C or C++}

`random div 141 mod 2` {in Pascal; see note below}

yields 0, Jane calls first; otherwise John calls first. Whenever Jane calls first, print "Snap! for Jane: " followed by Jane's face-up pile, from top to bottom. Whenever John calls first, print "Snap! for John: " followed by John's face-up pile. If the game ends, print "John wins." or "Jane wins.", whichever is appropriate. If the game does not end when each player has turned over 1000 cards, print "Keeps going and going ..."

### Sample input

```
ABCD A
CBADC
```

### Output for Sample Input

```
Snap! for Jane: BCBA
Snap! for Jane: DADCBCBA
Snap! for John: CBAC
Snap! for John: ADADCBAC
John wins.
```