

UCF “Practice” Local Contest — Aug 24, 2013

The Suffix Game

filename: suffixgame

Dr. Orooji’s twins, Mack and Zack, have discovered a new game to help them pass the time. First, Mack will select a word from the dictionary and write it down. Next, Zack selects a word and writes it down below Mack’s word, taking care that the ends of the words are aligned. For instance, if Mack chooses the word "random", and Zack chooses "kingdom", then they will write them down as:

```
random
kingdom
```

Observe that the shorter word has been padded with a space at the front, to make the ends of the words line up.

Next, they will take turns examining the last letter of each word. If both strings end with the same letter, it will be removed from the ends of both strings. This procedure continues until the final characters do not match. So, for the previous example, the procedure will give:

```
random  -->  rando  -->  rand  -->  ran
kingdom      kingdo      kingd      king
```

There is a catch, though. When the game ends, the two resulting words are written down in a special notebook. Since the twins insist on writing *exactly* two words, neither word must be completely consumed. Thus, the shortening procedure will end if either of the strings is shrunk down to a single character (see Sample Input/Output for clarification).

The Problem:

Dr. Orooji has decided to beat the twins at their own game, literally. Your job is to assist him by writing a program that, given the two starting words of a game, will play the game to the end and output the two final words.

The Input:

Mack and Zack play multiple games. The first line of the input to your program will be an integer g , indicating the number of games to be played. This will be followed by g lines, with two words on each line, indicating the two initial words. The words will consist of only lowercase letters ('a' – 'z'), each word at least one letter and at most 20 letters, first word starting in column one and the second word separated from the first word by exactly one space.

The Output:

At the beginning of each test case, output "*Game #g:*", where *g* is the game number (starting from 1). Then, output the two input words (indented three columns). Finally, output the results of the game (indented three columns) by saying: "The words entered in the notebook are *w1* and *w2*.", where *w1* and *w2* represent the two final words, in the same order as the originals.

Leave a blank line after the output for each test case. Follow the format illustrated in Sample Output.

Sample Input:

```
4
random kingdom
win twin
crushed sofa
myucf ucf
```

Sample Output:

```
Game #1:
  The input words are random and kingdom.
  The words entered in the notebook are ran and king.

Game #2:
  The input words are win and twin.
  The words entered in the notebook are w and tw.

Game #3:
  The input words are crushed and sofa.
  The words entered in the notebook are crushed and sofa.

Game #4:
  The input words are myucf and ucf.
  The words entered in the notebook are myu and u.
```