# UCF "Practice" Local Contest — Aug 23, 2014

## Wheel of Universally Copious Fortune

filename: `copious`
(Difficulty Level: `Easy`)

In the game "Wheel of Fortune", the number of letters in a word is given and the contestants guess the letters in the word and, as some letters appear, the contestants guess the word. But, you are a computer scientist and know that you can write a program to search a dictionary and provide candidate words (possible matches) for you.

**The Problem:**

Given the dictionary and a partially-defined word, you are to determine the candidate words. Note that there may be no candidate words for a given partially-defined word.

**The Input:**

The first input line contains an integer n *(1 ≤ n ≤ 100)*, indicating the number of words in the dictionary. The dictionary words will be on the following n input lines, one word per line. Each word starts in column 1, contains only lowercase letters, and will be 1-20 letters (inclusive). Assume that the dictionary words are distinct, i.e., no duplicates. The next input line will contain a positive integer m, indicating the number of words to be checked against the dictionary. These words will be on the following m input lines, one word per line. Each word starts in column 1, contains only lowercase letters and hyphens, and will be 1-20 characters (inclusive). A letter in a position indicates that the word must have that letter in that position; a hyphen in a position indicates that any letter can be in that position.

**The Output:**

At the beginning of each word to be checked, output "`Word #`*w*`:`", where *w* is the word number (starting from 1). Then print the input word to be checked. Then, on the following output lines, print the candidate words from dictionary that could be a match (print these words in the order they appear in the dictionary). Also print the total number of candidate words (possible matches).

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

(Sample Input/Output on the next page)

**Sample Input:**

```
8
at
cat
ali
sat
nerds
coach
couch
ninja
5
co-ch
-at
---
ali
a-c
```

**Sample Output:**

```
Word #1: co-ch
coach
couch
Total number of candidate words = 2

Word #2: -at
cat
sat
Total number of candidate words = 2

Word #3: ---
cat
ali
sat
Total number of candidate words = 3

Word #4: ali
ali
Total number of candidate words = 1

Word #5: a-c
Total number of candidate words = 0
```