UCF Local Contest — September 1, 2012

Spaces, the Final Frontier

filename: `spaces`

Captain Pick-Hard, the commanding officer of the starship *Venture,* is also a skilled xenoarchaeologist. On his few vacations, he can generally be found on some obscure planet digging up artifacts from long dead civilizations. On his last expedition, he found a series of inscriptions in the ancient language of Zagziggian. He requires your help in deciphering them.

A strange feature of Zagziggian writing is that it doesn't use spaces between words. Consequently, deciphering it is generally quite difficult, as one can never be sure when one word ends and the next word begins. This makes division of an inscription into its component words a difficult problem, because there can be a very large number of possible divisions.

Fortunately, the ever helpful Commander Theta has successfully reactivated the Zagziggian Codex, an artifact that, given an inscription, assigns a numerical value to a set of Zagziggian words. After studying these word-value pairs, Captain Pick-Hard hypothesizes that the *score* of a division is the sum of the values of its component words. And furthermore, inscriptions are written in such a way that the correct division is the one with the highest score. Note that the same word may occur multiple times in an inscription, and each occurrence contributes to the score. So if the word *ra* has a score of 5, and shows up 3 times, that counts as 5*3 = 15.

The Problem:

The captain wants a program that will find the highest scoring division of each inscription. Helping the captain is a step towards the UCF Programming Team and all the glory that goes with it, so let's help the captain.

The Input:

The first input line contains a positive integer, *n*, indicating the number of inscriptions to divide. This is followed by *n* descriptions of an inscription and the associated Codex.

The first line of each description is a single string *S* with length between 1 and 1000, inclusive. This string is guaranteed to contain only lowercase letters with no spaces.

The next input line contains an integer, c $(1 \leq c \leq 1000)$, indicating the number of word-value pairs generated by the Codex for this inscription. This is followed by *c* lines, each of which contains a word *w* (all lowercase letters with length between 1 and 1000, inclusive) and its value v $(1 \leq v \leq 1000)$, separated by a single space.

The Output:

For each of the descriptions in the input, output "`Inscription #k:`" where *k* is the number of the inscription. The next output line should contain the highest possible score for dividing that inscription. On the next line, you should print the highest scoring division by inserting spaces into the inscription string at the appropriate spots. You may assume that at least one valid division always exists.

In case there are multiple possible divisions with the highest score, choose the one where the first *differing* space is placed earlier in the string. (A differing space is one that is not present in both divisions.) See the Sample Output for any clarifications.

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

Sample Input:

```
2
helloworld
7
he 5
hello 9
llo 2
lo 3
wo 6
wor 4
world 8
atatata
5
a 7
at 3
ta 6
t 4
att 5
```

Sample Output:

```
Inscription #1:
17
hello world

Inscription #2:
40
a t a t a t a
```