Re-Exam Practice Problems, Continued

18. A **prefix atom** of a list L of strings is a string A in L such that, for any string W in L, if W is a prefix of A, then W is equal to A.

First write a function **`is_prefix_atom(L,A)`** that returns **`True`** if and only if A is a prefix atom of L. Then write a function **`prefix_atoms(L)`** that returns a list of the distinct prefix atoms of L sorted by increasing length.

Next write a function **`prefix_dictionary(L)`** that returns a dictionary whose keys are the prefix atoms of L; and the value in the dictionary for prefix atom A is a list of the distinct strings in L having A as a prefix.

Complete your program by prompting the user for a string and printing, for each prefix atom of the word list of S, the atom, A , the character : and then the list of words of S having A as a prefix. Each atom and list should be on a separate line.

Note that u is a prefix of v iff **`v.startswith(u)`** returns **`True`**

Example run on next page, with user input underlined:

```
Enter a string: This is not an instance of Isner's best effort
but his serve is a button on his coat of achievements and has
served him well; notably, it could be in the list of the best in
history

a : ['a', 'an', 'achievements', 'and']

is : ['is', "isner's"]

it : ['it']

in : ['in', 'instance']

be : ['be', 'best']

on : ['on']

of : ['of']

has : ['has']

his : ['his', 'history']

but : ['but', 'button']

not : ['not', 'notably,']

him : ['him']

the : ['the']

this : ['this']

list : ['list']

coat : ['coat']

well; : ['well;']

serve : ['serve', 'served']

could : ['could']

effort : ['effort']
```