PROBLEM 4: WORD SEARCH

For this exercise you must write a program that helps the user with a word search. The user will input the grid of letters, followed by a list of words to find in the grid. Your program must then output each word from the list in the same order given, followed by the coordinates of each letter in that word.

Your program will receive input as soon as it launches. The first section of input will consist of one or more lines of uppercase letters representing the grid of letters to be searched. This section of input will be followed by a hyphen ("-") on a line by itself. The second section of input will consist of one or more words to be found in the grid, each on its own line. This section of input will also be followed by a hyphen ("-") on a line by itself. Your program must then output each word from the given list in the order given, followed by a colon (":"), followed by the coordinates of each letter in the word, expressed as row number, comma, column number surrounded by parantheses (e.g. `(12,13)`). Each coordinate must be separated from the others by a space. The words may be forwards or backwards, and may be horizontal or vertical; you will not need to account for diagonal words. The list of valid words must be output in the same order as the words were originally given and the coordinates must be in word order (i.e. the first coordinate must correspond to the first letter of the word, the second coordinate to the second letter, and so on). If a given word cannot be found in the grid, your program must output "Not found." in place of the list of coordinates.

You may safely assume that each line of input will consist of no more than 1,000 characters, that all characters will be encoded as valid UTF-8, and that your program will be given the locale `en_US.utf8`. You may safely assume that all input will be upper case English letters without diacritical marks. Each word will appear only once in the grid of letters.

*Your program's output must match the examples given below **exactly**. Your program must not prompt for input. Note carefully the spelling, capitalization, punctuation, and spacing of the output. The input that will be given to your program is highlighted in **red**.*

EXAMPLE RUN 1

```
XXXXXXXXXX
XXXXTSAXXX
XXXXXXXXXX
-
TSA
CODING
-
TSA: (2,5) (2,6) (2,7)
CODING: Not found.
```

EXAMPLE RUN 2

```
GXXXXXXXXX
NXXXTSAXXX
IXXXXXXXXX
DXXXXXXXXX
OXXXXXXXXX
CXXXXXXXXX
-
TSA
CODING
-
TSA: (2,5) (2,6) (2,7)
CODING: (6,1) (5,1) (4,1) (3,1) (2,1) (1,1)
```

EXAMPLE RUN 3

```
MWPALWEAMU
LBAUEZMDJM
NBYCGHJONI
NOITAVONNI
QFPSCMWWCB
ICHOYOQZJG
-
LEGACY
OF
INNOVATION
-
LEGACY: (1,5) (2,5) (3,5) (4,5) (5,5) (6,5)
OF: (4,2) (5,2)
INNOVATION: (4,10) (4,9) (4,8) (4,7) (4,6) (4,5) (4,4) (4,3) (4,2) (4,1)
```