

Problem H: Rock Skipping

Source file: `rock.{c, cpp, java}`

Input file: `rock.in`

As a member of the International Rock-Skipping League, you travel to a different lake each week and compete in a rock-skipping contest. The goal is to throw a rock so that it skips as many times as possible; the exact rules for determining the winner are given below. To make the competitions interesting, the IRSL often chooses lakes with logs, sandbars, and other obstacles. You are provided with a side-view, water-level "map" of the lake as shown in the top line of the example below. (The numbers 0..29 below the map are just for reference.) A period (".") indicates clear water, where a rock will skip; any other character indicates some kind of obstacle that will stop a rock.

```
...=...**..#...@.....:.....=. .  
-----  
111111111112222222222  
012345678901234567890123456789
```

You stand at the left end of the lake. You can throw a rock so that it lands at any position in the lake, and then skips at any fixed interval thereafter. So a throw can be defined as a pair (i,d) , where $i \geq 0$ is the initial landing position and $d > 0$ is the distance between skips. Note that d must be positive. The *count* of a throw is the number of times that it skips on the water. The *length* is the position of its last contact with either the water or an obstacle. To rank two distinct throws, use the following criteria, in order, until a winner is determined: count (highest wins); length (greatest wins); initial position (greatest wins); distance between skips (smallest wins).

For the map shown above, throw $(27,2)$ hits the obstacle at position 27; it has count 0 and length 27. Throw $(16,1)$ skips at positions 16, 17, 18, and 19, then hits the obstacle at position 20; it has count 4 and length 20, so it beats throw $(27,2)$. Throw $(2,7)$ skips at positions 2, 9, 16, and 23, then skips over the lake; it has count 4 and length 23, so it beats throw $(16,1)$. Throw $(1,4)$ skips at positions 1, 5, 9, 13, 17, 21, 25, and 29, then skips over the lake; it has count 8 and distance 29, and is the best possible throw for this lake.

Input: The input consists of one or more lake maps, each 1–40 characters long and on a line by itself, followed by a line containing only "END" that signals the end of the input. Positions within a map are numbered starting with zero. Maps will only contain printable ASCII punctuation characters. A period indicates clear water and any other character indicates an obstacle.

Output: For each map, compute the best possible throw (i,d) , then output a line containing i and d separated by one space.

Example input:	Example output:
<pre>...=...**..#...@.....:.....=. . .(+ /^\. *++&* END</pre>	<pre>1 4 0 3 3 1 4 1</pre>

Last modified on October 26, 2007 at 9:28 AM.