QUEE MAR

Filename: `marquee`

Glenn's Sign Company is expanding into new markets. Previously, they focused entirely on fixed signs and hired graphical artists to do their work. Glenn is now interested in expanding his company into electronic signs – that is, marquees.

A marquee is a sign where a programmed message scrolls across an electronic display, allowing for longer messages than fixed billboards. Since this is all new to Glenn's Sign Company, they're hiring lots of new programmers and artists to work on these new signs.

The Problem:
The signs have a message that is to be scrolled, as well as a fixed number of characters that they can display at any one time. You are to simulate one full cycle of the marquee showing each part of the message as it would be displayed for each movement until it wraps around. A movement is defined as the leftmost character disappearing from the display, the remaining characters shifting to the left by one slot, and the next character appearing on the rightmost slot.

The Input:
The first line of the input will contain a single positive integer, n, representing the number of signs in the file. The next $2n$ lines will represent the individual signs. The first line of each sign will contain a string, s (of between 1 and 99 characters, inclusive), representing the message that will be programmed into the sign. The message will contain only uppercase and lowercase letters as well as spaces. There will be no leading or trailing spaces on the line (that is, the first and last character on the line will be an uppercase or lowercase letter). The second line of each sign will contain a single integer, m ($1 \leq m \leq 50$), representing the number of characters the sign can display at any one time.

The Output:
For each sign, the output should first contain the sign header, "`Sign #`*n*`:`", on a line by itself where *n* is the number of the sign (beginning with 1). Next, each sign should contain, in sequential order, the series of displays (each on a line by itself) the marquee should have using square brackets ('`[`' and '`]`') to show where the edges of the sign are. Each sign should be displayed for one full cycle, starting with the first character of the message in the leftmost slot on the sign. There should be a blank line separating the output for each sign. If *m* is greater than or equal to the length of *s*, then the sign does not need to cycle and only the original message should be displayed left-justified (use spaces in the output for unused sign characters). When the marquee is wrapping around, there should be exactly one space inserted between the last character and first character of the message. All other spacing should be preserved exactly as it appears in the input file.

Sample Input:
```
Computer
10
ACM
2
```

Sample Output:
```
Sign #2:
[Computer  ]

ACM
2
Sign #1:
[AC]
[CM]
[M ]
[ A]
```