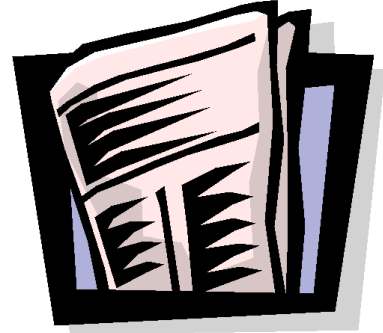


## Problem B: Raggedy, Raggedy

Consider the problem of laying out text in lines of a fixed maximum width  $L$  (a.k.a., “line filling”).

If you do a poor job, the ends of the lines are unnecessarily ragged - like this paragraph. Now, by convention, we allow the last line of a paragraph to be arbitrarily ragged. We don’t mind if that final line contains just a few characters, but we expect the earlier lines to be of approximately uniform length, filling up the column in which we are setting the text.



The straightforward approach of filling each line with as many words as will fit and then moving to the next line does not always yield the most aesthetically pleasing results. For example, the sequence

See if we care.

could be laid out in  $L=6$  like this:

```
See if
we
care.
```

That layout is, arguably, not as visually pleasing as

```
See
if we
care.
```

Define a “word” as any sequence of non-whitespace characters bounded by a line start or end or by a blank. The legal “*whitespace characters*” in this problem are blanks and the line terminator characters.

Given a sequence of  $N$  words of width  $w_1, w_2, \dots, w_N$ , and a maximum line width  $L$ , with the guarantee that for all  $i$ ,  $w_i \leq L$ , define  $w(i, j)$  as the width of the line containing words  $i$  through  $j$ , inclusive, plus one blank space between each pair of words.

Then we can define the raggedness of a line containing words  $i$  through  $j$  as

$$r(i, j) = (L - w(i, j))^2$$

Write a program to read paragraphs of text and to lay them out in a way that no line contains more than  $L$  characters, for a specified  $L$ , and so that you minimize the total raggedness added up over all lines except the last one. (The final line of a paragraph can be arbitrarily shorter than the lines above it.) Line terminator characters are not counted as part of the line width.

## Problem B: Raggedy, Raggedy

### Input

Input will consist of one or more datasets.

Each dataset begins with a line containing one integer,  $L$ , denoting the maximum line width (not counting line terminator characters). You are guaranteed that  $0 < L \leq 80$ . A value of zero indicates the end of input.

The remainder of the dataset consists of up to 250 lines containing a paragraph of text, terminated by an empty line. Paragraphs may contain from 1 to 500 words, where a word is any consecutive sequence of non-whitespace characters.

No line of text will contain a word of length greater than  $L$ .

### Output

Print each paragraph laid out optimally as described above. After each paragraph print a line containing “===” (three equal signs).

If there is more than one way to fill a paragraph with the optimal raggedness, any such layout may be printed.

### Example

Given the input

```
6
See if we
care.

25
Raggedy, raggedy are we.
Just as raggedy as raggedy can be.
We don't get nothin' for our labor.
So raggedy, raggedy are we.
- P Seeger

0
```

the output would be

```
See
if we
care.
===
Raggedy, raggedy are
we. Just as raggedy
as raggedy can be. We
don't get nothin' for
```

## Problem B: Raggedy, Raggedy

---

our labor. So raggedy,  
raggedy are we. - P  
Seeger  
===