CS 280: Program #1

Fall 2016

Due September 30 at 11:55PM, via Moodle

A text formatter is a program that arranges text according to rules specified by the user of the formatter. A web browser acts as a text formatter, as does a word processing tool like Microsoft Word. For this assignment you will implement a small amount of text formatting functionality. You will read in text and reformat it according to instructions given to your program.

The input to your program consists of lines of text. The text consists of a sequence of words (characters) separated by whitespace (spaces, tabs, newlines). The output of your program is a reformatting of the input according to specified rules.

The output of your program consists of lines of text of a specific length. The text should be right justified to the specified line length.

Deleted: , almost all of which should have the same length

For example, if the input of your program is as follows:

```
Screen door slams. Mary's
dress
waves. Like a     vision she dances
across the porch as the
radio      plays.
```

Deleted: sways

If you are attempting to generate 40 character lines, the output would be:

Deleted: asked

```
Screen  door slams.  Mary's dress waves.
Like _a _vision _she _dances _across the
porch as the radio plays.
```

Deleted:

Deleted: sways

Deleted:

Deleted: ¶

Observe the following from this simple example:

- Words on the input might be delimited by more than one instance of whitespace. The program treats multiple spaces on input as a single space.
- In order to make the output lines fit the proper length, extra spaces may need to be added between words on output.
- The last line is less than the specified number of characters, because there are no more words left.

Deleted: . The extra spaces are added in multiple places

Some additional specifications:

- The default line length is 60 characters. The minimum line length is 10 and the maximum is 120.
- If spaces must be added between words in order to pad the length of the line, the following rules should be observed:
  - Spaces should be added equally between all words in a line. The position of an added space can be decided randomly; however the number of spaces added between all the words should differ by at most one space.
  - It is unacceptable to add more than three spaces between two words; the program should add an extra word instead.
  - The last line in a paragraph does not need to have any spaces added to pad out its width.
  - In the case of a single word on a line other than the last line in a paragraph, up to two spaces may be placed before the word when generating output.

Formatted

- One or more blank lines is taken to mean “end of paragraph”; when it reads a blank line, your program should end the current paragraph and skip a line. Several consecutive blank lines should be treated as a single blank line: only one line should be skipped between the paragraphs.
- Reaching the end of input should cause the paragraph in progress to be completed; however, an additional blank line should NOT be generated at the end of the output.
- There may be a circumstance where it is impossible to generate a line less than the specified length (for example, suppose that the line is of length 10 and a word is of length 15). In those cases, your program should insert a dash character to break the long word into two pieces, and should put the second piece at the beginning of the next line. It is acceptable to perform this operation multiple times on a single word.
- A line in the form “.ll N”, where N is a positive integer greater than or equal to 10 and less than 120, causes a new paragraph to begin. All subsequent lines will be of length N. If N is not a positive integer that is less than 120, or if there are any extra words at the end of the line, the entire line is ignored and a new paragraph does NOT begin.
- If a previous paragraph has ended and the first line of the next paragraph has not yet been generated, then additional blank lines and/or .ll directives should cause no additional lines to be skipped before beginning the next paragraph.
- Any other line that begins with a dot is ignored.
- Ignoring lines that begin with a dot that are not .ll directives, or ignoring .ll lines that have any errors, is done silently.

Your program will take a single command line argument, which is the name of the input file to read. The output should be written to standard out.

Deleted: .