1. The biggest obstacle I encountered was figuring out how to find the words in the input file. I knew that writing a while function of getting character would help, but I had no idea how to individually pass down the words I found. I tried to find what distinguishes a word from another word, and I finally realized that spaces tell me when a word ends and another begins. I set up a c string and put the characters my loop gets into the string and that is my word. I then figured out that I need to break out of the loop when the program hits a space, so a single word is extracted out from the input file without any spaces. I decided to process one word at a time, so when I print it out, I also empty the word c string in order to store the next word in the input.
2. First check if the input for len is valid,

If not, return 2;

Repeatedly, look for a word and process it:

Repeatedly, get a character from the input file:

If it is not a space, store it in the word c string;

If it is a space, break out of the loop and process the word

If the input file has nothing in it, return 0 and print nothing;

If the word is longer than the desired len value, split it on different lines and return 1;

Check if the word is a paragraph break;

If it is a repeated paragraph break, ignore it;

If it is the first one, make a paragraph break;

If the word is not a paragraph break and the line is empty:

print it directly and store it into the line c string to keep track of the size of the line;

Else if the word ends in “.” or “?”:

If adding it and two spaces would not exceed the len value, print the word and store it into the line c string;

If adding it and two spaces would exceed the len value, print the word in a new line and reset the line c string to a newline with the word;

Else if the word is not empty:

If adding it and one space would not exceed the len value, print the word and store it into the line c string;

Else if adding it and one space would exceed the len value, print the word in a new line and reset the line c string to a newline with the word;

Empty the word c string for the next round;

If the end of file is reached, end the output with a newline and break out of the loop because everything behaved correctly;

Return 0 if everything behaved correctly;

1. 1. Two spaces after “.” and “?”: len = 40; Hello. How are you? Goodbye.

2. One space after other words: Hello. len = 40; How are you? Goodbye.

3. Word longer than line length: len = 15; The intermediate is called Fructose-6-Phosphate.

4. Process only one paragraph break: len = 10; Hello #P# #P# #P# Helloooo!

5. Nothing in the input file, should return 0: len = 10; (Nothing in the input)

6. #P##P# is regarded as a word: len = 30; Hello #P##P# Helloooo!

7. Desired line lengths is less than 1, should return 2: len = 0; Hello #P##P# Helloooo!

8. The last output line ends with a new line: len =10; Hello. How are you? Goodbye.

9. Handle longer inputs: len = 40; It always does seem to me that I am doing more work than I should do. It is not that I object to the work, mind you; I like work: it fascinates me. I can sit and look at it for hours. I love to keep it by me: the idea of getting rid

of it nearly breaks my heart. #P# You cannot give me too much work; to accumulate work has almost become

a passion with me: my study is so full of it now, that there is hardly an inch of room for any more.