CS417 Lab 6

This lab is meant to get you started reading text files, and also working with text formatting. Your goal is to reflow the text in a file: you will print the text, inserting line breaks, so that the printed text is not too wide.

Getting Started

To begin the lab:

- Create a folder, in your `Documents` folder.
- Go to `mycourses.unh.edu`, click on the `unh.box.com` link, find the lab, and locate the program `reflow.py`. Download it into the folder that you just created. Also, download the file `independence.py.`
- Open the file `independence.txt`. Notice that it contains several lines of text. Each line has a different length.

Your Tasks

Edit the program `reflow.py`, and make these changes:

1. (5 points) Implement `print_file`, as follows:

   ```
   handle = open(filename, 'r')
   lines = handle.readlines()
   handle.close()
   print (lines)
   ```

   Notice that `lines` is a list of strings. Each string is line from the file. Also, notice that each string ends with `'\n'`.

---

2. (10 points) **FIRST**: copy the body of `print_file` into `print_lines`. Create a `for`-loop:

   ```
   for line in lines:
   ...
   ```

   inside the `for`-loop, put `print (line)`

   Notice that the lines are printed, but with a blank line each time. This is due

to the `"\n"` at the end of each line: `print()` adds another `"\n"`, so you get two `"\n"` per line.

Remove that `"\n"`. You can do this in several ways:

```
line = line[:-1]
```

or

```
line = line.rstrip('\n\r')
```

The second choice is better: it works with all operating systems, regardless of the newline character they might use.

---

3. (5 points) **FIRST**: copy the body of `print_lines` into `print_splits`.

   Instead of `print (line)`, write

   ```
   words_in_line = line.split()
   print (words_in_line)
   ```

   This shows that `split()` creates a list of strings, one for each word in the line.

---

4. (15 points) **FIRST**, copy the body of `print_splits` into `print_words`. Replace `print (words)` by a `for`-loop, that prints each `word` in `words_in_line`.

   IMPORTANT: print the words on a single line, something like this:

   ```
   print (word, end=' ')
   ```

   At the very end of the function, add a single `print()` (at left-most indentation)

   WHY?

   You'll get one single long line of words, each separated by a space.

---

5. (10 points) How long is this output? **FIRST**, copy the body of `print_words` into `words_length`. At its top, initialize `line_length` to 0. Instead of `print (word, end=' ')` update `line_length`:

```
    line_length += len(word) + 1
```

(+ 1 because there is ONE space after each word). At the very end, put `print (line_length)`

That's a VERY long line! We need to break it up.

---

6. (35 points) **FIRST**, copy the body of `words_length` into `print_reflowed`. Before printing the next word, we'll check if that would overflow the line width. If so, we'll `print()` to start a new line.

   Do the following for each word:

   if `line_length + 1 + len(word)` is bigger than `width`, then do a single `print()`, and reset `line_length` to zero.

   then, *after the* `if`, just print the word (using `end=" "`), and update `line_length`.

---

7. (Worth the last 20 points) Implement `print_right_aligned`. A right-aligned paragraph has spaces added at the beginning of each line, as in this example:

```
  When in the Course of human events it
    becomes necessary for one people to
dissolve the political bands which have
     connected them with another and to
  assume among the powers of the earth,
the separate and equal station to which
 the Laws of Nature and of Nature's God
```

   The problem is that you don't know how many spaces to add, until you have gathered all the words in the line.

   I suggest you do the following:

   a. create a list of saved words, for each line.

   b. this list is initially `[]`

   c. when the line length is about to exceed `width`, you should
      1. print exactly the right number of spaces: `width - line_length`
      2. print each of the words in the saved list
      3. `print()`
      4. reset the saved list to `[]`

   d. where you used to do `print (word, end=' ')`, simply append the word

to the saved list.

e. At the very end, the saved list will still have some words in it! Don't forget to print them too!

---

8. (Bonus 10 points) Implement `print_justified`, which inserts extra spaces *between* words, just enough to create a perfect rectangle of text:

```
When  in  the Course of human events it
becomes  necessary  for  one  people to
dissolve the political bands which have
connected  them  with  another  and  to
assume  among  the powers of the earth,
the separate and equal station to which
the  Laws of Nature and of Nature's God
```

Notice the first line: there are *two* extra spaces between the first few words, but a single space thereafter. You'll need a counter, which initially stores the # of extra spaces needed. As you print each word, decrement the counter. When it's zero, stop decrementing.

Turning in your work

When you are done, go to `mycourses.unh.edu`, and find the cs417, and find the lab. Click the `Submit` button, and upload `reflow.py`. You should turn in your work at the end of the lab session, even if you haven't completed it. You will have until midnight to turn in your work again without a lateness penalty.