Software Engineering 265
Software Development Methods
Summer 2014

Assignment 2

Due: Friday, June 27, 11:30 am by submission via Subversion
(no late submissions accepted)

Programming environment

For this assignment you must ensure your work executes correctly on the Linux
machines in ELW B215 (i.e., these have Python 3.3 installed). You are free to do
some of your programming on your own computers; if you do this, give yourself a
few days before the due date to iron out any bugs in the Python script you have
uploaded to the BSEng machines.

Individual work

This assignment is to be completed by each individual student (i.e., no group work).
Naturally you will want to discuss aspects of the problem with fellow students, and
such discussion is encouraged. **However, sharing of code fragments is strictly
forbidden without the express written permission of the course instructor
(Zastre).** If you are still unsure regarding what is permitted or have other questions
about what constitutes appropriate collaboration, please contact me as soon as
possible. (Code-similarity analysis tools may be used to examine submitted
programs.)

Objectives of this assignment

- Learn to use basic features of the Python 3 language. (To enable this in your
  environment when logged into the lab, type in the following command at the
  start of your session at the shell prompt: `scl enable python33 bash`
- Use the Python programming language to write a less resource-restricted
  implementation of "sengfmt" (but without using regular expressions).
- Use subversion to manage changes in your source code and annotate the
  evolution of your solution with "messages" provided during commits.
- Test your code against the twenty provided test cases.

This assignment: "sengfmt.py"

You are to write a Python version of the C program prepared for assignment #1. All of the formatting capabilities of that C program are to be implemented in this new program. Like the C program, it must run as a command in the shell. However, there are several additional abilities your Python script must have:

- If a filename is provided to the script, then that file's contents are formatted. If no filename is provided, then the contents of stdin are formatted. In either case, the formatted output is sent to stdout.

- The "?width", "?mrgn" and "?fmt" commands may appear at any line of the input file. As in assignment #1, such commands are valid only if that appear at the start of a line.

- The margin may now be specified relative to the value of the current margin position. For example, if the margin currently is 10, then when "?mrgn +5" is encountered the margin would be set to 15. If the margin currently is 20, then "?mrgn -7" would set the margin to 13, and a further "?mrgn -7" would set the margin to 6.

- Valid margins must be (a) greater than or equal to 0, and (b) less than or equal to the pagewidth minus 20. If a margin would be negative after a "?mrgn" command, then the margin is set to zero. If a margin would be greater than then pagewidth minus 20, then the margin is set to pagewidth minus 20.

- For this assignment, a "?mrgn" will never be immediately followed (i.e., on the next line) by another "?mrgn" command.

- There is no limit to the number of input lines. You may assume that each input line is no longer than 132 characters.

With your completed "sengfmt.py" script, the input would be transformed into the output (here redirected to a file) and then checked:

```
% ./sengfmt.py /home/zastre/seng265/assign2/in11.txt > ./myout11.txt

% diff /home/zastre/seng265/assign2/out11.txt ./myout11.txt
```

where the file "myout11.txt" would be found in your current directory.

Exercises for this assignment

1. Within your Subversion project create an "assign2" subdirectory. Use the test files in /home/zastre/seng265/assign2. (Files in01.txt through to in10.txt

are the same as those used for the first assignment.) Your "sengfmt.py" script must appear here. Ensure the subdirectory and script file are added to Subversion control.

2. Write your program. Amongst other tasks you will need to:
   - read text input from a file, line by line
   - write output to the terminal
   - extract substrings from lines produced when reading a file
   - create and use lists in a non-trivial array
   - use the "diff" Unix command to test the output of your program

3. Keep all of your code in one file for this assignment. In later assignments we will use the multiple-module features of Python.

4. Keep all of your code in one file for this assignment.

5. Use the test files to guide your implementation effort. Start with simple cases (such as those given in this writeup). Refrain from writing the program all at once, and budget time to anticipate when "things go wrong".

6. For this assignment you can assume all test inputs will be well-formed (i.e., our teaching assistant will not test your submission for handling of input or for arguments containing errors). Later assignments will specify error-handling as part of the assignment.

7. Reasonable run-time performance of your script is expected. None of the test cases should take longer than 15 seconds to complete on a machine in ELW B215.

What you must submit

- A single Python script file named "sengfmt.py" within your subversion repository containing a solution to Assignment #2.

Evaluation

Students will demonstrate their work to a member of the course's teaching team. Sign-up sheets for demos will be provided a few days before the due-date; each demo will require from 10 to 15 minutes.

Our grading scheme is relatively simple.

- "A" grade: An exceptional submission demonstrating creativity and initiative. "sengfmt" runs without any problems. The program is clearly written and uses functions appropriately (i.e., is well structured).
- "B" grade: A submission completing the requirements of the assignment. "sengfmt" runs without any problems. The program is clearly written.
- "C" grade: A submission completing most of the requirements of the assignment. "sengfmt" runs with some problems.
- "D" grade: A serious attempt at completing requirements for the assignment. "sengfmt" runs with quite a few problems.
- "F" grade: Either no submission given, or submission represents very little work.