Assignment #16

Implementing Linux Commands in Python

The goal of this assignment is to implement Python versions of certain Linux commands.

To see the specifications of each such *command*, type, in a terminal:

`man` *command*

The Python versions need not support any command line *options*, apart from those given below. In each case, the output format of the Python version should match that of the Linux version. If any file cannot be read, then the Python version should behave just like the Linux version; apart from that, no further error-checking need be done.

Implement *at least three* of the following Python programs, corresponding to Linux commands of the same name but with the first letter written in lower case:

1. `Wc`
   Assume that at least one filename argument is supplied, and output each number in a field of width 5.

2. `Uniq`
   Assume that either one or two filename arguments are supplied.

3. `Tail`
   The option '`-n`' should be supported. Assume that exactly one filename argument is supplied. For space efficiency, avoid storing the entire contents of the file in memory; for time efficiency, avoid reading the file twice.

4. `Grep`
   The options '`-n`' and '`-v`' should both be supported, but the *pattern* being searched for should just be a single string. Assume that at least one filename argument is supplied.

5. `Factor`
   Assume that all the command-line arguments or terminal inputs are non-negative integers, and note that 0 and 1 are considered to have no prime factors.

Program Submission:

Store the solutions in files named '`Wc`', '`Uniq`', '`Tail`', '`Grep`', and '`Factor`', respectively, and turn them in for grading by typing:

```
submit-cs1117 Wc
submit-cs1117 Uniq
submit-cs1117 Tail
submit-cs1117 Grep
submit-cs1117 Factor
```

Due Date: Fri Feb 12, 10:30am