Unix Tools	**Homework 2**
CS114 Fall 2002	**Due Date: Wednesday, October 16, 2002 (at noon)**

You should do all the work yourself, without any assistance from other people. If you need help, post your question to *cornell.class.cs114* newsgroup or if you want to keep your question private, e-mail me at *cs114@cs.cornell.edu*.

I will not accept late submissions.

Setup

1. Create a subdirectory `cs114-HW2` in your home directory. Make sure you are the only one who has any kind of access permissions to it.

2. Go to ∼`cs114/HW2` directory. It has a subdirectory set up for each of the CS114 students. Copy all the files from ∼`cs114/HW2/`*`yournetid`* to your `cs114-HW2` directory. (In case you are wondering - these files contain HTML pages with results of google.com searches).

Part I: Emacs

For this part, use Emacs (which you can invoke using `emacs`). Some of these problems are quite tricky, so triple-check your results. You may find the `diff` utility useful for checking what happened.

1. In `file1`, replace the first character of every line by `###` followed by a space in lines 10 through the end of the file. For example, you would change the line

   ```
   xyz
   ```

   into

   ```
   ### yz
   ```

   Hint: Look at the Emacs command `replace-regexp` (with `M-x describe-function`). It takes a regular expression and something to replace it with. It replaces all occurrences of the regular expression in the file, from the point where the cursor is. How do you position the cursor to line 10? (Look at the mode line carefully.) Note that a line can be longer than what fits on the screen, and thus may wraparound. How can you recognize the beginning of a line with a regular expression?

2. In `file2`, replace each digit between 1 and 5 (inclusive) with two symbols `[]`.
   Example: `12ab8rs5dA` should become `[][]ab8rs[]dA`.

3. In `file3`, change all the lines that contain at least one `|` symbols in the following way: replace the **first** `|` by `-|-` Try doing this using a single `replace-regexp` command.
   Example: `qWe|rtY|uio` should become `qWe-|-rtY|uio`
   Hint: You will need to look at some documentation about regular expressions to answer this problem (for instance, any of the links I provide on the course web page), and experiment a little bit. More precisely, try to figure out how to use `\(`, `\)`, `\1`, `\2`, etc. What does the regular expression `\(xy\)\1` match? What does `[^abc]` match?

Part II: Grep

1. Using the `man` command on `babbage`, read the `grep` man page and find out, how to tell it to count the number of lines matched.

2. Create a file named `answer` in your `cs114-HW2` directory.

3. Count the number of lines in `file4` that have at least 3 occurences of some 2-character string (each line can have its own 2-character string). Put this number on the first line of the `answer` file, followed on the same line by the `grep` command you issued to get that result.
   Example: `qw1-wt1-aaass1-xce` should be counted because `1-` appears 3 times.
   Hint: Recall the regular expression constructs you had to use question 3 above.

4. Same as the previous question, but when the first of the two symbols in that string is not a lower-case `a`. Write the number and the command used in the second line of the `answer` file.

5. In the `file5`, count the number of lines that either

   - has a lower-case letter that goes after `c` in the alphabet, then at least two of `a`, `b`, or `c`, then another lower-case letter that goes after `c` in the alphabet. **Example:** `qaabw`
   - or has three `a`’s in a row (`aaa`).
   - or there is an `n` immediately after every `a`. **Hint:** this includes lines with no `a`’s at all.

   Put this number on the third line of the `answer` file, followed again by the command you issued.
   Warning: none of the versions of `egrep` installed on `babbage` recognizes the `{...}` construction.

To submit your HW, run ∼`cs114/bin/submit-hw2`. The script will work as many times as you run it, but **you will only receive credit for your first submission**. If something goes wrong, please let me know ASAP.