Unix Tools	**Homework 2**
CS114 Fall 2003	**Due Date: Wednesday, October 22, 2003 (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 every cs114 student. 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).
3. Create a file named `answer` in your `cs114-HW2` directory.

Part I: 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. Count the number of lines in `search.html` that have at least an occurence of some 3-character string (each line can have its own 3-character string) that starts with a digit and ends with a digit. 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: `qw1-3t1-aaass1-xce` should be counted because `3t1` appears in the line.
3. Same as the previous question, but when the middle symbol in the string is not a lower-case `a`. Write the number and the command used in the second line of the `answer` file.
4. In the `search.html`, 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.

Part II: Crawling over HTML code

As most of you know, most pages on the web are coded using a fairly simple markup language called HTML, that essentially specify how things should be presented to the user. For more info on HTML, see the web classic
http://archive.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html
for a quick introduction.

Here is a very cut down sample HTML file:

```
<html>
  <head>
    <title>Content of /home/sample</title>
  </head>
  <body>
    <ul>
      <li><a href="schedule.html">schedule.html</a></li>
      <li><a href="papers/index.html">papers/</a></li>
      <li><a href="software/index.html">software/</a></li>
    </ul>
  </body>
</html>
```

A few things are worth noticing about the above: (1) the title of the page is contained with `<title>` and `</title>` tags, (2) an HTML link is of the form `<li><a href="path or URL">some text</a></li>`. Roughly speaking, a link in HTML is interpreted as follows: when you select the text between <*a*> and </*a*> on the web page, the browser loads the file or web page given by *href*.

In this part of the homework, you will write commands to crawl over HTML code (as returned by the google search engine) to extract useful bits of text embedded in the text. For the google results, we will extract the search strings, the number of hits, the name of the links, and the links themselves.

You can use Emacs to look at the .html files to get a feel for what they contain. You can also use `lynx` to display the web page, e.g., `lynx search.html`.

The commands you write should read from an HTML file and print to stdout some result. What do I mean by command? Any pipeline of Unix utilities. For example, here is a command to take `search.html` and print all the lines containing `<title>` in it, in sorted order: `grep ’<title>’ search.html | sort`. That should give you an idea what I want.

You can use any utility that processes text. You may want to especially look at `sed`, most importantly, its man page.

1. As I said, each of the files you copied in your directory was the HTML result of a google search. Write a command to extract the string searched for in a google result pages. For example, it takes `search.html` and returns the search that corresponding to that page. As a hint, notice that the search strings are given in the title of the web page. Here’s the kind of output I want:

   ```
   babbage% some command looking at search.html
   uncertainty
   ```

   Remember, you can pipeline together arbitrary commands. Put the command you use in the fourth line of the `answer` file.

2. Write a command to extract the total number of results returned by the search. This information appears in the form “Results 1 - 10 of about 2,240,000”. In that case, for example, I want you to return 2,240,000. Put the command you use on the fifth line of the `answer` file.

3. This is slightly more complex, but involves the same ideas as in the last two questions. Write a command to extract all the names of the result of a search. Consider an example. In my case, my `search.html` file contains the following result (among others):

   ```
   <a href=http://www.afrc.af\.mil/afrcpubs/pubs/rp.htm>Recurring
   ```

```
Periodicals</a><br><font size=-1> <b>...</b>
<b>RP65</b>-<b>1</b>. Three Times a year. AFRC Financial Management
Bulletin. FM.<br>90--Command Policy. RP90-<b>1</b>. Quarterly. AFRC
IG Crossfeed Newsletter.<br>IG. <b>...</b> <br><font
color=#008000>www.afrc.af.mil/afrcpubs/pubs/rp.htm -  8k - </font><a
class=fl
href=http://216.239.39.100/search?q=cache:W0NEgFYf83UC:www.afrc.af.mil/
afrcpubs/pubs/rp.htm+rp65+1&hl=en&ie=UTF-8>Cached</a> - <a class=fl
href=/search?hl=en&lr=&ie=UTF-8&q=related:www.afrc.af.mil/afrcpubs/pubs/rp.htm>
Similar pages</a></font> <p>
```

(Wouldn't it be nice if Google returned pretty-printed HTML? Sigh.)

A result in google will (typically!) start with a link to the page, and contain some extract of the page, etc. It (typically!) ends with a link to "Similar pages". You can use that sort of information to delimit a result. There are (typically!) ten results per page, to give you an idea. For this question, you should return the list of all the names of results on the page: the name of the result above is "Recurring Periodicals". (The name is contained within the *<a>* and *</a>* tags.). As I said, there should be ten of them per page. Put the command you use on the sixth line of the `answer` file.

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.