System Programming

Regular Expressions
UNIX programs that use REs

- grep (search within files)
- egrep (grep with extended RE’s)
- vi/emacs (text editors)
- ex (line editor)
- sed (stream editor)
- awk (pattern scanning language)
- perl (scripting language)
Basic vs. Extended REs

- In basic regular expressions the metacharacters ?, +, {, }, (, ), |, and ) have no special meaning (grep)
  - To give them special meaning, use the escaped versions: \?, \+, \{, \}, \(, \), and \|

- When using extended regular expressions, these metacharacters have special meaning
  - `grep -E = egrep`
Using egrep

- `egrep pattern filename(s)`
- **To be safe, put quotation marks around your pattern**
- **Examples:**
  - `egrep "abc" textfile`  
    (print lines containing “abc”)
  - `egrep -i "abc" textfile`  
    (same, but ignore case)
  - `egrep -v "abc" textfile`  
    (print lines not containing “abc”)
  - `egrep -n "abc" textfile`  
    (include line numbers)
  - `egrep -c "abc" textfile`  
    (print a count of lines containing “abc”)
Metacharacters

- Period ( . ): matches any single character
  - “a . c” matches abc, adc, a&c, a;c, …
  - “u . . x” matches unix, uvax, u3(x, …)

- Asterisk ( * ): matches zero or more occurrences of the previous RE
  - not the same as wildcards in the shell!
  - “ab* c” matches ac, abc, abbc, abbbbc, …
  - “. *” matches any string
Plus ( + ): matches one or more occurrences of the preceding RE
- “ab+c” matches abc, abbc, but not ac

Question mark ( ? ): matches zero or one occurrence of the preceding RE
- “ab?c” matches ac, abc but not abbc

Logical or ( | ): matches RE before or RE after bar
- “abc | def” matches abc or def
Metacharacters (cont.)

- Caret ( ^ ): means beginning of line
  - “^D.*” matches a line beginning with D

- Dollar sign ( $ ) means end of line
  - “.*d$” matches a line ending with d

- Backslash ( \ ): escapes other metacharacters
  - “file\.txt” matches file.txt but not file_txt
Square brackets ([ ]): specifies a set of characters as a list
- any character in the set will match
- ^ before the set negates the set
- – specifies a character range

Examples:
- “[fF]un” matches fun, Fun
- “b[aeiou]g” matches bag, beg, big, bog, bug
- “[A-Z].*” matches a string starting with a capital letter
- “[^abc].*” matches any string not starting with a, b, or c
Metacharacters (cont.)

- Parentheses ( () ): used for grouping
  - “a(bc)*” matches a, abc, abcbbc, abcbbcbbc, ...
  - “(foot|base)ball” matches football or baseball

- Braces ( {} ): specify the number of repetitions of an RE
  - “[a-z]{3}” matches three lowercase letters
  - “m.{2,4}” matches strings with m followed by between 2 and 4 characters
What do these mean?

- **Examples**
  - `egrep "^B.*s$" file`
  - `egrep "[0-9]{3}" file`
  - `egrep "num(ber)? [0-9]+" file`
  - `egrep "word" file | wc -l`
  - `egrep "[A-Z].*\?" file`
  - `ls -l | egrep "^....r.-r.-"`

- **What if grep was used instead?**
  - Search for users with user IDs containing at least two 0s
    - `grep "^[^:]*:[^:]*[^:]*0[^:]*0[^:]*:.*" /etc/passwd`

- **/etc/passwd file format**
  - `<username>:x:<userid>:<groupid>:<useridinfo>:<homedir>:<loginshell>`
  - An x character indicates that encrypted password is stored in `/etc/shadow` file
Word searching with egrep

- The system may have a small dictionary for checking spelling: /usr/dict/words
- Find words that contain all five vowels in alphabetical order
- `cat alphvowels`
  
  `^[^aeiou]*a[^aeiou]*e[^aeiou]*i[^aeiou]*o[^aeiou]*u[^aeiou]*$`
- `egrep -f alphvowels /usr/dict/words`
  
  affectious
  facetious
  ...

Find all words of six or more letters that have the letters in alphabetical order.

```
cat monotonic
```

```
egrep -f monotonic /usr/dict/words | grep "......."
abdest
almost
biopsy
...
```
Practice

- Construct `egrep` commands that find in file:
  - Lines beginning with a word of at least 10 characters
  - Lines containing a student ID number in standard 3-part form
  - Number of lines with 2 consecutive capitalized words
  - Number of lines not ending in an alphabetic character
  - Lines containing a word beginning with a vowel at the end of a sentence