CS 246
C and Unix
Exam 1

Name	

Question	Points	Score
1	21	
2	21	
3	6	
4	12	
5	30	
6	10	
Total:	100	

1. (21 points) Write a SINGLE glob expression that matches each set of files. Make it as simple as possible.

   (a) Files with names ending in .png.

   Solution: `*.png`

   (b) Files with names beginning with alpha and ending with .py.

   Solution: `alpha*.py`

   (c) Files with names beginning with a letter (upper or lower case).

   Solution: `[a-zA-Z]*`

   (d) Files with names that end with either .html or .css.

   Solution: `*.{html,css}`

   (e) Files with names containing colonel.

   Solution: `*colonel*`

   (f) Files with names whose length is 8 characters ending in .txt.

   Solution: `????.txt`

   (g) Files with names consisting of numbers between 5000 and 5999 followed by .jpg.

   Solution: `5[0-9][0-9][0-9].jpg`

2. (21 points) Write a SINGLE regular expression that can be used with egrep with NO options to match each set of lines as described below. Make it as simple as possible.

   (a) Lines beginning with first and ending with last.

   Solution: `^first.*last$`

   (b) Lines consisting entirely of c's or d's (possibly empty).

   Solution: `^[cd]*$`

(c) Lines beginning with plugh that have at least 7 characters.

Solution: `^plugh..`

(d) Lines beginning with plugh that have at most 10 characters.

Solution: `^plugh.{,5}$` or `^plugh.?.?.?.?.?$` or

(e) Lines containing either xyzzy or plugh.

Solution: `xyzzy|plugh`

(f) Lines whose length is even.

Solution: `^(..)*$`

(g) Lines that contain no digits (0-9) characters.

Solution: `^[^0-9]*$`

3. (6 points) Give the octal permissions that result in each symbolic permission.

   (a) `r---w---x`

   Solution: `421`

   (b) `rw-r-x-wx`

   Solution: `653`

4. (12 points) Write a unix command or combination of commands connected by pipelines that does each of the following. You may not use intermediate files. There may be several different ways to do each one.

   (a) Reads from standard input and writes to standard output, replacing a's by 0's, b's by 1's, and c's by 2's.

   Solution: `tr abc 012`

   (b) Writes to standard output the first 10 lines that begin with an uppercase character of a file called plugh.

   Solution: `egrep '^[A-Z]' plugh | head`

   (c) Writes to standard output the contents of a text file named xyzzy, with the lines sorted, with no duplicates.

   Solution: `sort xyzzy | uniq`

   (d) Prints the count of the blank lines in a file called plugh.

   Solution: `egrep '^$' plugh | wc -l`

5. (30 points) Matching. Write the number of the best answer in each blank.

(a) __6__ The key sequence for exiting emacs.

(b) __11__ The person who originally designed and implemented C.

(c) __8__ The data structure that contains file permissions.

(d) __5__ The modifier key that is used for most commands that apply to characters in emacs or bash.

(e) __3__ A command that can be used to replace an entire word with a different word in a file.

(f) __2__ The current working directory.

(g) __13__ A command that can replace or delete characters.

(h) __12__ The home directory.

(i) __15__ The person that developed Linux.

(j) __16__ The key sequence for deleting to the end of the line in emacs or bask.

(k) __9__ The parent of the current working directory.

(l) __1__ The modifier key that is used for most commands that apply to entire words in emacs or bash.

(m) __14__ The person who originally implemented Unix.

(n) __7__ The person who developed Gnu emacs.

(o) __17__ Determines the permissions for files when they are first created.

1. meta
2. .
3. sed
4. Steve Jobs
5. ctrl
6. ctrl-x ctrl-c
7. Richard Stallman
8. inode
9. ..
10. Bill Gates
11. Dennis Ritchie
12. ~
13. tr
14. Ken Thompson
15. Linus Torvalds
16. ctrl-k
17. umask
18. none of the above

6. (10 points) A data file has 5 columns of numbers separated by spaces. Write an AWK program that prints out, for each line, the sum of the numbers in columns 1 and 3, along with the total of the individual sums at the bottom.

Solution:

```
{
  print $1 + $3;
  total += $1 + $3;
}

END {
  print total;
}
```