1. Describe three different ways of setting the permissions on a file or directory to `r--r--r--`. Create a file and see if this works.
2. Team up with a partner. Copy `/bin/sh` to your home directory. Type "`chmod +s sh`". Check the permissions on `sh` in the directory listing. Now ask your partner to change into your home directory and run the program `./sh`. Ask them to run the `id` command. What's happened? Your partner can type `exit` to return to their shell.
3. What would happen if the system administrator created a `sh` file in this way? Why is it sometimes necessary for a system administrator to use this feature using programs other than `sh`?
4. Delete `sh` from your home directory (or at least to do a `chmod -s sh`).
5. Modify the permissions on your home directory to make it completely private. Check that your partner can't access your directory. Now put the permissions back to how they were.
6. Type `umask 000` and then create a file called `world.txt` containing the words "`hello world`". Look at the permissions on the file. What's happened? Now type `umask 022` and create a file called `world2.txt`. When might this feature be useful?
7. Create a file called "`hello.txt`" in your home directory using the command `cat -u > hello.txt`. Ask your partner to change into your home directory and run `tail -f hello.txt`. Now type several lines into `hello.txt`. What appears on your partner's screen?
8. Use `find` to display the names of all files in the `/home` subdirectory tree. Can you do this without displaying errors for files you can't read?
9. Use `find` to display the names of all files in the system that are bigger than 1MB.
10. Use `find` and `file` to display all files in the `/home` subdirectory tree, as well as a guess at what sort of a file they are. Do this in two different ways.
11. Use `grep` to isolate the line in `/etc/passwd` that contains your login details.
12. Use `find` and `grep` and `sort` to display a sorted list of all files in the `/home` subdirectory tree that contain the word `hello` somewhere inside them.
13. Use `locate` to find all filenames that contain the word `emacs`. Can you combine this with `grep` to avoid displaying all filenames containing the word `lib`?
14. Create a file containing some lines that you think would match the regular expression: `(^[0-9]{1,5}[a-zA-z ]+$)|none` and some lines that you think would not match. Use `egrep` to see if your intuition is correct.

15. Archive the contents of your home directory (including any subdirectories) using `tar` and `cpio`. Compress the `tar` archive with `compress`, and the `cpio` archive with `gzip`. Now extract their contents.
16. On Linux systems, the file `/dev/urandom` is a constantly generated random stream of characters. Can you use this file with `od` to printout a random decimal number?
17. Type `mount` (with no parameters) and try to interpret the output.