Exercises:

1. What are the permissions for newcatfile.txt? Change these permissions to read-only (by owner, group, and world). Now try to delete the file. What happens? Change the permissions back. Now delete the file. This is a simple way to make sure that you don't accidentally delete important files (for example, the raw data from your sequencing run). (hint: `chmod`)

`ls -l` to see permissions.
change to read only:
`chmod 444 newcatfile.txt`

2. What does the gzip utility do? (look at the manual page). Use gzip to compress one of the files in your directory. gzip is fantastically effective for biological sequences, as these files have low information content to begin with. Use gunzip to uncompress the file. How much memory does the file occupy, uncompressed? how big is the compressed file? (hint: `ls -lh`)

>gzip is a compression utility and will create a .gz version of any file that you give it. gunzip will uncompress a .gz file.

3. You can use the old and very effective archive utility, `tar`, ('tape archive') to compress a directory into a single file. It is convenient to transfer sets of files this way as they are kept organized, and compression reduces the risk of file corruption in transit. Look at the man page for `tar`. Create a directory and create some emtpy files in it, by using the commands `mkdir` and `touch`. Use `tar` to create an archive from the directory and its contents. Now, extract the directory and remake the archive, this time with compression. (hint: `tar -cf, tar -czf, tar -xf` etc)

`tar -czf archive.tar.gz tempdirectory file1 file2`
this will create a file, named archive.tar.gz, that contains tempdirectory and two files, and that is compressed with gzip.
To extract the archive:
`tar -xzf archive.tar.gz`

Note that like most Unix utilities, tar is not careful about clobbering files, so if you unpack an archive full of files in your directory and you already had files with those names (e.g. README), they will be overwritten.

>in fact, I usually create a new directory, move the tar archive into it, and extract it in there to avoid these issues.

4. Create a file called chrnames, that contains these lines:

chr2
chr5
chr7

The script below will create a new blank file for each line in this file. Run it, to check.

```
#!/bin/bash

for i in `cat chrnames`
do
   echo $i
   touch $i.peaks.bed
done
```

Modify the script so that it reads the file, and for each line in the file, it should create a new file containing all lines in chip_peaks.bed that match the chromosome specified.

```
#!/bin/bash

for i in `cat chrnames`
do
   echo $i
   grep $i chip_peaks.bed > $i.peaks.bed
done
```

5. On a Unix platform, compiling programs is a common task. We will download and compile samtools (a very popular toolkit for manipulating alignment data). Download the source release from here: http://www.htslib.org/download/
and follow the instructions to compile and install it. A convenient place to install this is in /usr/local/bin/ (if you are the administrator of the computer). If you are not the administrator, simply install it into any folder that you own.

You may run into trouble . . . if you see an error, try typing `which gcc` and see what you get. If you are using a Mac, you may not have a compiler installed, in which case you'll need to get the Developer Toolkit.

>Here you just need to follow the directions to use the configure and make utilites with the scripts provided with the samtools package.