This assignment assumes you have read Chapters 2, 3, 4 and 5 of Sobell. Some specific cross-references may also be given in the questions below.

Prepare your answers to the questions below in a single plain ASCII text file[1]. Submissions in other formats will be ignored. If you work with a partner, make sure the submitted file contains a properly-completed copy of the partners form posted on the assignments page. Failure to do that will result in at least one of you not receiving credit for the assignment.

Submit your file to the Curator system, under the heading `L02`, by the posted deadline for this assignment. No late submissions will be accepted.

For questions 1-5, assume there is a directory on my CentOS virtual machine, `submissions`, that contains files as shown in the long listing (`ls -l`) below:

```
#1013 wmcquain@centosVM submissions> ls -l
total 1100
-rw-rw----. 1 wmcquain wmcquain  960 Feb 23 10:17 aaron03.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 2406 Feb 23 10:17 abdel98.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 2027 Feb 23 10:17 ajay99.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 1468 Feb 23 10:17 ajd29.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 1654 Feb 23 10:17 ajorq21.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 1462 Feb 23 10:17 alexf98.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 1419 Feb 23 10:17 alexfrigault.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 1535 Feb 23 10:17 alexmill.prog01.2.c
-rw-rw----. 1 wmcquain wmcquain 2060 Feb 23 10:17 aliadams.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 2088 Feb 23 10:17 aligy.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 2008 Feb 23 10:17 alyss17.prog01.2.c
-rw-rw----. 1 wmcquain wmcquain 1102 Feb 23 10:17 amandal5.prog01.1.c
. . .
-rw-rw----. 1 wmcquain wmcquain 1345 Feb 23 10:17 zankvon.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 1379 Feb 23 10:17 zhangtr.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 1917 Feb 23 10:17 zhewei6.prog01.2.c
-rw-rw----. 1 wmcquain wmcquain 1458 Feb 23 10:17 zicong7.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 2145 Feb 23 10:17 zoromskik.prog01.1.c
-rw-rw----. 1 wmcquain wmcquain 2434 Feb 23 10:17 zruijie8.prog01.1.c
```

The directory contains over 1000 files, and you do not know all the file names. You do know that all of the file names take the following form

```
PID.prog01.X.c
```

where `PID` indicates the student who submitted the file, and `X` is a single digit, called the *submission number*.

You may benefit from consulting the `man` page for the `ls` command.

1. **[24 points]** Assume that my current working directory is `submissions`. Write a single Linux command that I could execute to display the names of the files described.

   a) All and only the files, if any, for which the PID field contains the letter `'m'`.

   b) All and only the files, if any, for which the PID field ends with the letter `'m'`.

   c) All and only the files, if any, for which the submission number is 2.

2. **[16 points]** Suppose my current working directory is `submissions`. For each part, give a single valid Linux command that I could use to create the specified `tar` archive.

   a) Bundle all the files in the current working directory whose names have a PID field beginning with the letter `'r'` into a flat tar file named `r_files.tar`, so that the tar file is created in my home directory.

   b) Bundle all the files with submission number 3 into a flat tar file named `third_subs.tar`, so that the tar file is in the parent of the current working directory.

3. **[12 points]** Repeat question 2b, but this time, create a zip archive instead of a tar file; naming the zip file `third_subs.zip`.

4. **[16 points]** Now, suppose that my current working directory is the parent directory for the directory `submissions`. Suppose that I want to list the names of all the files in the `submissions` directory for which the PID field contains the character `'g'`.

   a) Why might the following command not produce the output I want? A good explanation would include an example.

   ```
   ls --format=single-column ./submissions/*g*
   ```

   b) Why might the following command not produce the output I want? A good explanation would include an example.

   ```
   ls --format=single-column ./submissions/*g*.*
   ```

For question 5, suppose you current working directory is `J3Files`, which has the following contents, including a directory tree rooted in `CS3114`:

```
J3Files
|-- CS3114
|   `-- J3
|       |-- Client
|       |   `-- Point.java
|       `-- DS
|           |-- Compare2D.java
|           |-- Direction.java
|           |-- Lewis.class
|           `-- prQuadTree.java
|-- gradeJ3.sh
|-- LogComparator.jar
|-- PRGenerator.jar
`-- testDriver.java
```

5. **[32 points]** For each part, write a single Linux command that will achieve the specified action, with no unnecessary side-effects, assuming the command will be executed in the directory `J3Files`. The parts are independent; in other words, each will be performed with the system in the state shown above. You will find a number of `man` pages to be very useful. There will be no partial credit for these answers.

   a) Copy the file `testDriver.java` to the user's `~/temp` directory.
   b) List the files in the subdirectory `DS`, sorted in ascending order by file size (smallest files first).
   c) Copy the entire directory tree rooted in `CS3114`, including all the files therein, to the user's `~/backup` directory.
   d) Remove the entire directory tree rooted in `J3`, including the directory `J3` itself.

[1] How can you tell if you've prepared a plain ASCII text file? The simplest way is to use a Linux text editor to create the file you are going to submit (e.g., gedit, geany, etc). Alternatively, use the `file` command in a Linux shell:

```
centOS > file Tolstoy.txt
Tolstoy.txt: ASCII text
```

It is possible you may get a different response; for example:

```
ASCII text, with very long lines
ASCII text, with CRLF line terminators
ASCII text, with very long lines, with CRLF line terminators
UTF-8 Unicode text
UTF-8 Unicode text, with very long lines
UTF-8 Unicode text, with CRLF line terminators
UTF-8 Unicode text, with very long lines, with CRLF line terminators
Non-ISO extended-ASCII text
Non-ISO extended-ASCII text, with very long lines, with CRLF line. . .
Pascal source, ASCII text
C source, ASCII text
```

Those are probably all acceptable. However:

- long lines indicate your file may be difficult to view in certain applications; try inserting line breaks around column 80 when you type your files
- CRLF line terminators indicate your file was produced on a Windows environment; that may also cause issues if your file is opened in certain applications
- Non-ISO extended ASCII text indicates you've used non-standard characters in your file; this may be due to the inclusion of garbage text you don't intend to submit; such characters may cause some applications to interpret the file incorrectly, or even to refuse to open it

On the other hand, responses like these from the `file` command would indicate your file is certainly not acceptable:

```
Microsoft Word 2007+
PDF document, version 1.5
Rich Text Format data, version 1, ANSI
POSIX tar archive
data
gzip compressed data, from Unix. . .
```

You can also use the `cat` or `less` command to display your file to a Linux terminal window. If this displays anything other than the simple text of your answers, it's not a plain text file.

Apple users beware: the standard text editors on Apple computers seem to save files in RTF format, even if they indicate they are saving as plain text. We will NOT grade RTF files. You must either install a text editor that can save files in the correct format, or prepare your file using a Linux editor.

If you aren't sure, ask a TA to look at your file before you submit it.