

Name: Gavin McRo

Question 1

```
total 0
42606069 -rw-r--r--  1 gavintaylormcroy  staff  0 Apr  5 08:33 file1.txt
42606070 -rw-r--r--  1 gavintaylormcroy  staff  0 Apr  5 08:33 file2.txt
Gavins-MacBook-Pro:activity gavintaylormcroy$
```

Command displayed inode number, permissions, time of creation, who created etc. The files also have two different inode values

Question 2

```
total 0
42606069 -rw-r--r--@ 1 gavintaylormcroy  staff  0 Apr  5 08:33 file1.txt
42606070 -rw-r--r--  1 gavintaylormcroy  staff  0 Apr  5 08:33 file2.txt
42612014 lrwxr-xr-x  1 gavintaylormcroy  staff  9 Apr  5 08:38 my_softlink -> file1.txt
```

The inode numbers are different for the soft link file. There is also a display to show my_softlink points to file1.txt.

After deleting file1.txt it no longer works because the original file it was pointing to is now missing. They also share different inode numbers, so the soft link file must've been separate from file1.txt, but still relied on file1.txt in order to behave as a link. This is because soft links refer to another filename in the system, so once the filename was gone, so was the link

Question 3

```
total 0
42606070 -rw-r--r--@ 2 gavintaylormcroy  staff  0 Apr  5 08:33 file2.txt
42606070 -rw-r--r--@ 2 gavintaylormcroy  staff  0 Apr  5 08:33 my_hardlink
42612014 lrwxr-xr-x  1 gavintaylormcroy  staff  9 Apr  5 08:38 my_softlink -> file1.txt
```

My_hardlink shares the same inode values as file2.txt, and it also doesn't seem to have a pointer to it. It's acting like it's the same file twice.

After deleting file2.txt the hard link still works. This is because the hard link points to the actual files inode, so deleting file1.txt just removes another link to the inode.