Part A:

1. Compile and run [3000test.c](http://homeostasis.scs.carleton.ca/~soma/os-2017f/code/tut3/3000test.c). It takes a filename as an argument and reports information on the file. Try giving it the following and see what it reports:
   * a regular text file that exists
     1. inode 400318, length of characters in text file 9, letter a counts in file 3
   * a directory
     1. inode 400319, length 4096
   * a symbolic link
     1. inode 400318, length of characters in text file 9, letter a counts in file 3
        1. same as regular txt file
   * a device file (character or block)
     1. inode 408, length 80
2. Change 3000test to use lstat rather than stat. How does its behavior change?
   * For symbolic link file
     1. Lstat() Inode is of the actual link file and doesn’t have a count
     2. While stat() output is identical to text file
3. Modify 3000test so when it is given a symbolic link it reports the name of the target. Use readlink(2).
   * Modified and needs argument passed to be a path instead of filename for symbolic link, and prints the target file name
4. Are there files or directories that you cannot run 3000test on? Can you configure file/directory permissions so as to make something inaccessible to 3000test?
   * I’m almost sure that you can run all files/directories on 3000test
   * You can if you add an if to check if it’s a regular file/directory with S\_ISREG or S\_ISDIR and check for file/directory name and if it is that fn then output a message saying inaccessible.
     1. This is implemented in line 53 of 3000test.c
5. How does the memory use of 3000test change as it runs? You may want to add calls to sleep(3) so you can observe its memory usage. You can create a 1 GB file of random data with the command dd if=/dev/urandom of=test bs=1024 count=1000000.
   * Only time there’s memory usage is when reading through data for a count in 3000test there’s 995584 memory used. The only code using memory is for loop of a count.

Part B:

1. Run ls -lai. What are the numbers appearing in the left column?
   1. The inodes of each file
2. Run dd if=/dev/zero of=foo bs=8192 count=32K What is the logical size of the file? How much space does it consume on disk? (Hint: Look at the size option to ls.)
   1. Logical size is 250 Mb, physical is 257 Mb \*use ls -ls first column is phys size
3. Run mkfs.ext4 foo. Does foo consume any more space?
   1. Doesn’t consume anymore space, still the same…….ext4 makes holes in file
4. Run dumpe2fs foo. What does the output of this command mean?
   1. prints the super block and blocks group information for the filesystem present on *device.*
5. What command do you run to check the filesystem in foo for errors?
   1. fsck foo is file system check
6. Run sudo mount foo /mnt. How does this command change what files are accessible?
   1. This makes so that the files accessible has to be through /mnt directory to get to foo
   2. Was allowed to access lost+found through /mnt
7. Run df. What device is mounted on /mnt? What is this device?
   1. /dev/loop2
      1. This device that makes the file accessible as block devices
         1. Typically used to mount disk images
8. Run chown student:student /mnt.
   1. Ran however operation isn’t permitted
9. Run rsync -a -v /etc /mnt. What does this command do? Explain the arguments as well. Did you get errors copying any files?
   1. Rsync is a tool for copying files and directories between local and remote systems
      1. Arguments are -a which is archive mode, that includes all necessary options like copy files recursively, preserving almost everything.
      2. -v is verbose that gives us info on what files are being transferred and a summary about the data transferred at the end.
   2. yes there are errors at the end, a bunch of files weren’t transferred
10. Run sudo umount /mnt. What files can you still access, and what have gone away?
    1. Cannot access lost+found through /mnt it went away
11. Run dd if=/dev/zero of=foo conv=notrunc count=10 bs=512. How does the "conv=notrunc" change dd's behavior (versus the command in question 2)?
    1. Notrunc prevents data deletion/truncation when copying with dd
12. Run sudo mount foo /mnt. What error do you get?
    1. Mount: /mnt: wrong fs type, bad option, bad superblock on /dev/loop2, missing codepage or helper program, or other error
13. What command can you run to make foo mountable again? What characteristic of the file system enables this command to work?
    1. Format the partition/create a new filesystem with mkfs.ext4 foo
14. Run the command truncate -s 1G bar. What is the logical size of bar, and how much space does it consume on disk? How does this compare with foo?
    1. Bar is 1Gb logical, 0 physical and foo is 250 Mb logical, 17 Mb physical
15. How does the logical size of bar change when you create an ext4 filesystem in it? What about the space consumed on disk?
    1. Logical size is still 1 Gb but physical is now 33 Mb

Part C:

1. Look at the hard link counts of files locally and compare those to the link counts over sshfs. How do they compare?
   1. 23 hard links locally ls -lai vs
2. Try running some of the sudo'd commands above (e.g., "sudo mount foo /mnt") without the sudo. Why do they fail? Try running them under strace and see how their system calls change. Do they figure out that they aren't running as root and abort, or do they try doing a privileged operation and it fails?
   1. Other doesn’t have permissions, since I gave it permissions it says
      1. For sudo
         1. /mnt special device foo doesn’t exist
      2. For non sudo
         1. /mnt must be superuser to use mount
3. Can you access sshfs mounted files as root? (You can become root by typing "sudo su -".) What happens?
   1. Permission denied as root still
   2. Need to sudo chmod 777 /mnt file to get perms, root still doesn’t work on sshfs
4. Look at inode numbers in local and remote filesystems (as reported by ls -i). How do they compare?
5. dd a large file to a local drive. Do same thing over sshfs. Which is faster? (What is a large file in this context?)
6. Can you sshfs to the SCS systems (e.g., access.scs.carleton.ca)?
7. How can you use the mount command to unmount a sshfs-mounted filesystem (rather than fusermount)?