Paul found a problem with question 10 -when you start gedit from a terminal with *sudo gedit*, save doesn't work properly in the latest Ubuntu.  There were several questions, so I thought I'd explain question 8 in detail.

10)  Now, let’s have some fun. Use sudo nano (or sudo gedit if you like) to create a file in /usr/local/bin with a filename “ls” and content like this:

*#!/bin/bash*  
*echo 'Ha!  Gotcha!  You thought this was ls!'*

Note:  use single quotes in the echo statement, not double quotes as in the assignment doc.  Single quotes stop BASH from expanding the content (variable names, special characters) and will stop errors caused by the exclamation point and double quotes.  See Double Quotes and then Single Quotes in The Linux Command Line, starting at page 76 in chapter 7.

Then make the file executable with  
*sudo chmod +x /usr/local/sbin/ls*

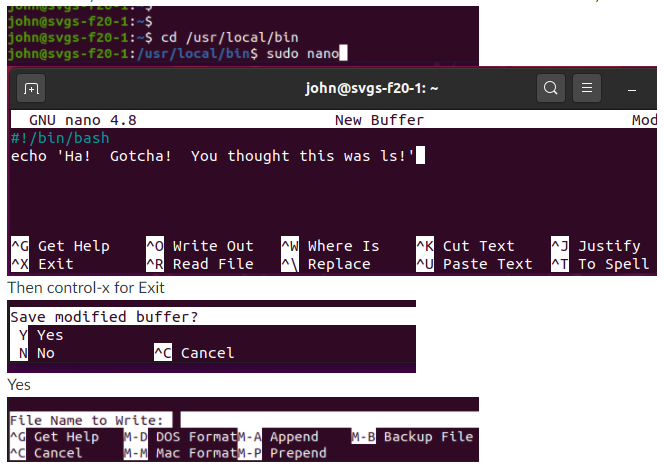
The purpose for this question:  
1)  show how the PATH environment variable works  
2)  show how PATH can be abused

The first step is to create a file with this content, and save it as /usr/local/bin/ls.  
#!/bin/bash  
echo 'Ha!  Gotcha!  You thought this was ls!'

You can make any text file executable if you list the program you want to use to execute the script in the file.  In this case, the first line *#!/bin/bash* tells the OS that when this file is executed, it should be executed using the BASH shell.

Here's how to create the file using nano

Another way is to use the nano editor.  You can also cd to the /usr/local/bin directory to make life easier.

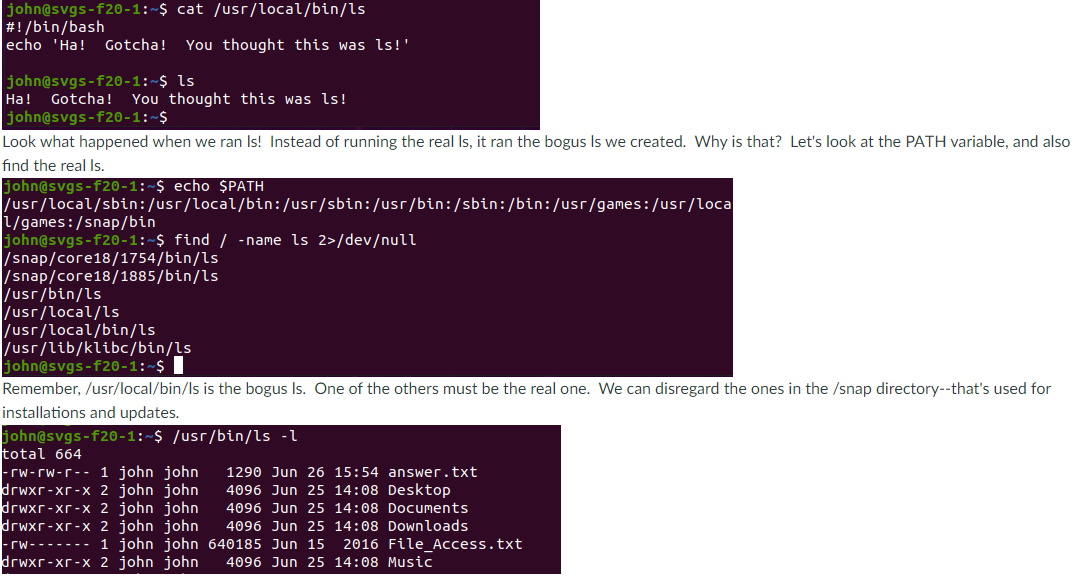


then type the file name, ls, and enter

If you want to be a Linux guru you can use the vi or vim text editor.

Don't forget:  
*sudo chmod +x /usr/local/sbin/ls*

At any rate, you should be able to cat /usr/local/bin/ls



Looks like /usr/bin/ls is the real one.  Now, why did the bogus ls run instead of the real one?  The answer is in the PATH.  If you look, /usr/local/bin (bogus ls) comes before /usr/bin (real ls) in the PATH.  As BASH checked the directories in PATH, it came across the bogus ls first, so that's what it ran.

To get your ls working again remove the bogus ls  
*sudo rm /usr/local/bin/ls*