OPS102 – Week 3 – File Systems

Student Name: Chetan Arora

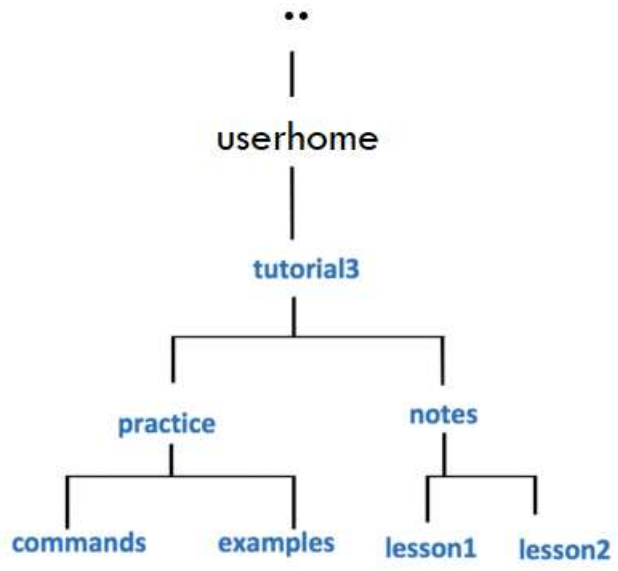
Student ID: 100976240

Activity 1: File Globing

When issuing Linux or Windows commands, it may be **more efficient** (less typing) to use **filename expansion symbols** also called **File Globing** to match files that share similar characteristics (e.g. same file extension) when issuing Linux commands.
***Example**:* You can use a special character to indicate to the Bash shell to match all files that end with the extension ".txt" in your current working directory:
ls *.txt
a.txt b.txt c.txt 1.txt 2.txt 3.txt abc.txt work.txt
Below are the most common Filename Expansion symbols and how they are used for filename expansion:

Filename Expansion Symbol	Purpose
*	Asterisk (*) to represent **0 or more characters**
?	Question mark (?) to represent **exactly one character (any character)**
[ ]	Square brackets ([ ]) to represent and match for the **character enclosed within the square brackets**. It represents ONLY ONE character - it's like a **Question Mark (?)** but with **conditions or restrictions.**
[!]	Square brackets containing an exclamation mark immediately after the open square bracket ([!]) to represent and match and **OPPOSITE** character for the character enclosed within the square brackets.

Consider following file hierarchy for the activities in this section. This applies to both of Linux and Windows.



You will now get practice issuing file management commands using **filename expansion symbols**. We will be using the directory structure given above.

A great way to practice filename expansion, use the **touch** command on Linux to create a lot of empty filenames (for windows use any preferred way to create such files.), write the **ls/dir** commands that use **filename expansion**, predict the filenames that will be display, and finally run the command to check your work.

Perform the following steps for Linux and repeat them for windows using equivalent commands learnt previously:

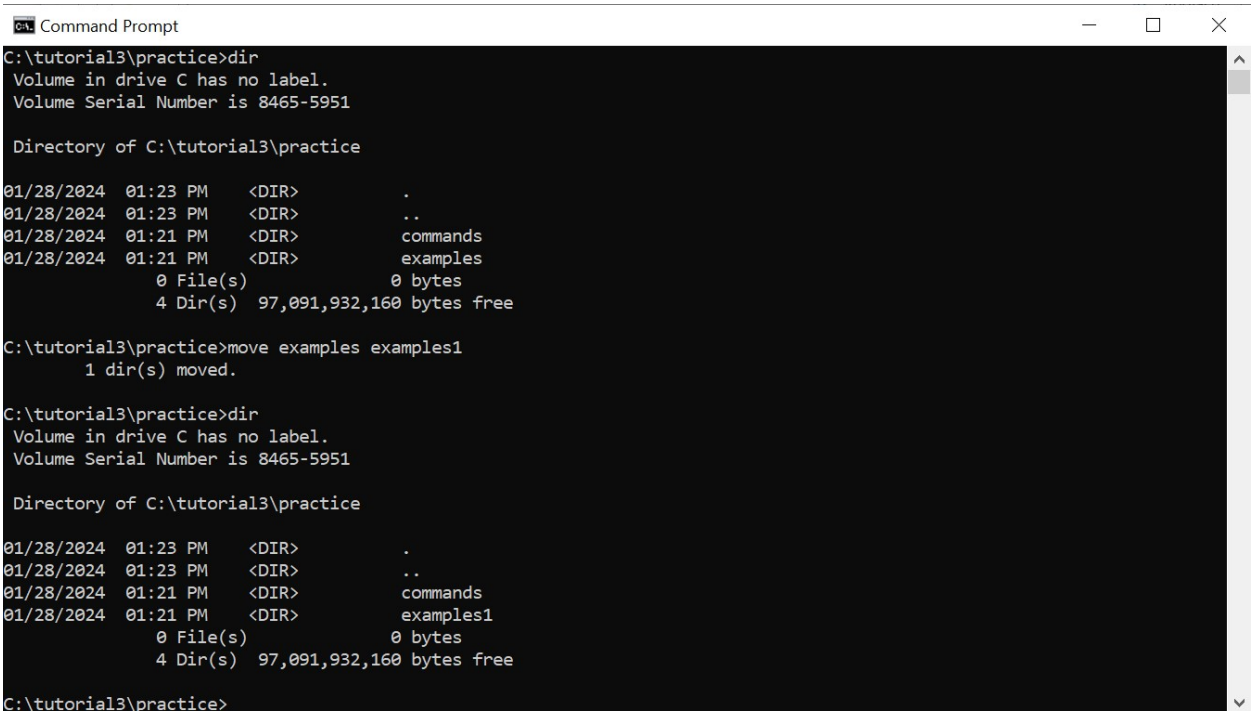
1. Issue a Linux command to move to the **examples** directory
   (i.e. under *practice* directory as shown in diagram to the right).

   Ans: Linux: - mv examples examples1

   carora18@mtrx-node03pd:~/tutorial3/practice

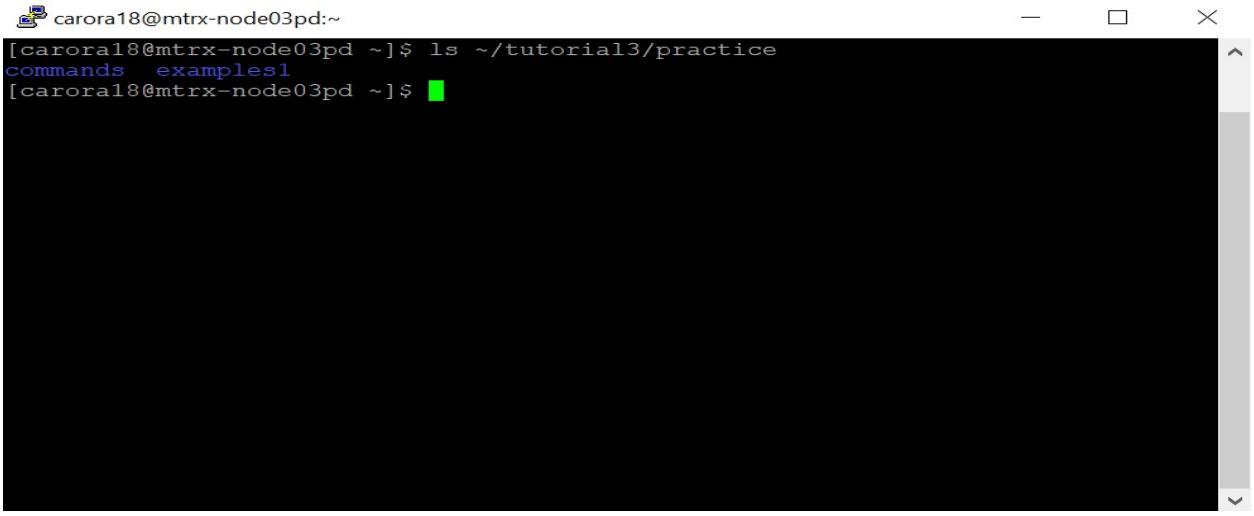
```
[carora18@mtrx-node03pd practice]$ mv examples examples1
[carora18@mtrx-node03pd practice]$ ls
commands  examples1
[carora18@mtrx-node03pd practice]$
```

   Windows: - xcopy /e /I examples examples1

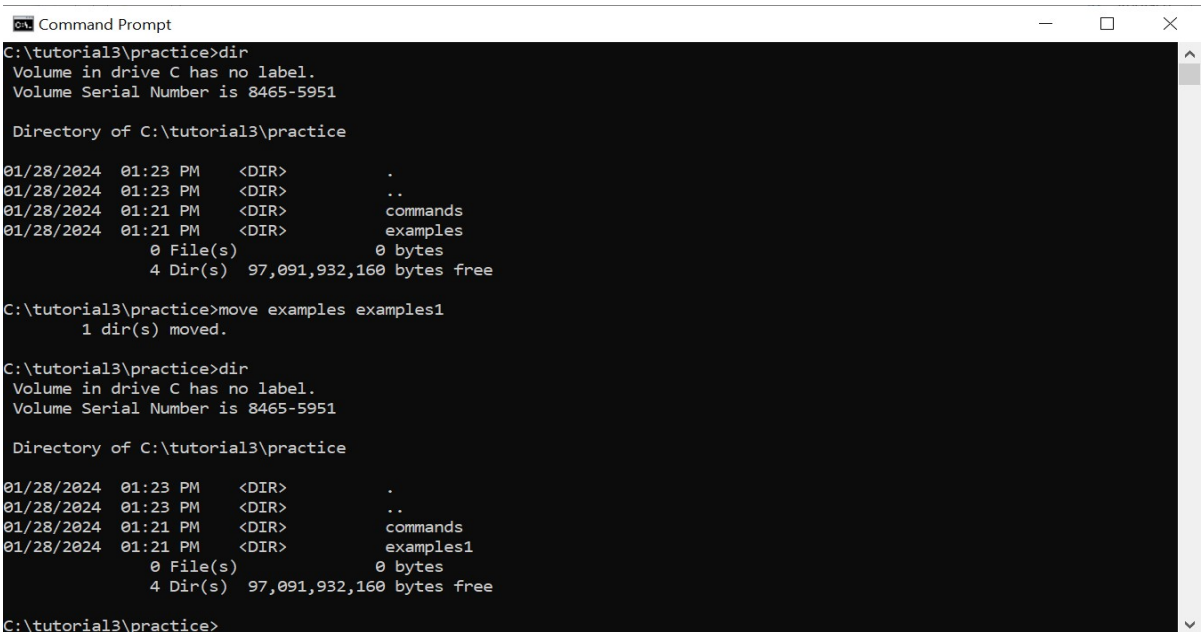


2. Issue a Linux command to confirmed that you have moved to the **examples** directory.

Ans: Linux: - ls ~/tutorial3/practice



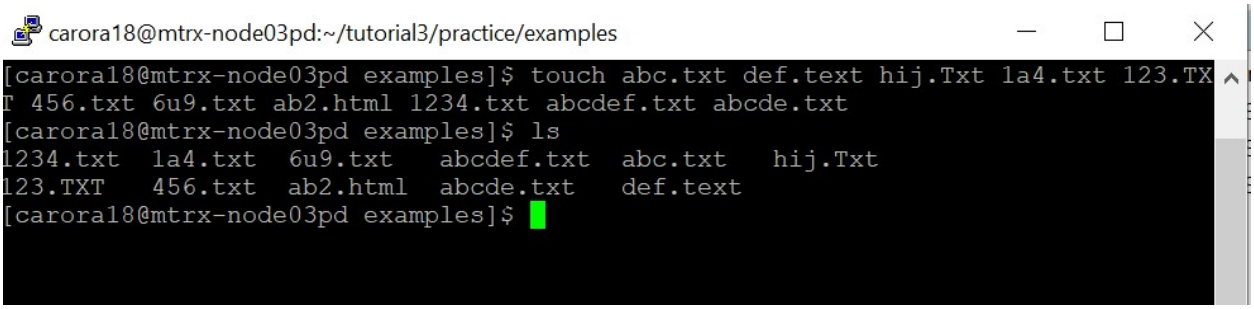
Windows: - dir tutorial3\practice



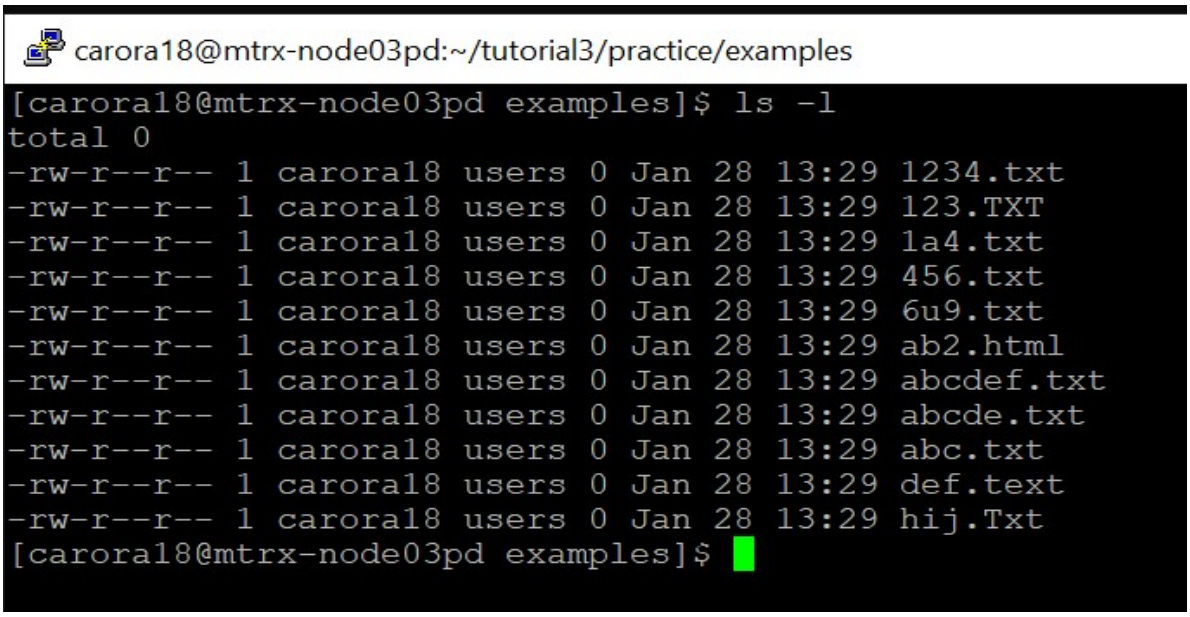
3. Issue the **touch** command to create the following empty text files in the *examples* directory:
   (note *upper* and *lowercase* letters)

abc.txt
def.text
hij.TxT
1a4.txt
123.TXT
456.txt
6u9.txt
ab2.html
1234.txt
abcdef.txt
abcde.txt

Ans: Change Directory to examples: - cd ~/tutorial3/practice/examples
Use Touch command: - touch abc.txt def.text hij.Txt 1a4.txt 123.TXT 456.txt 6u9.txt ab2.html 1234.txt abcdef.txt abcde.txt



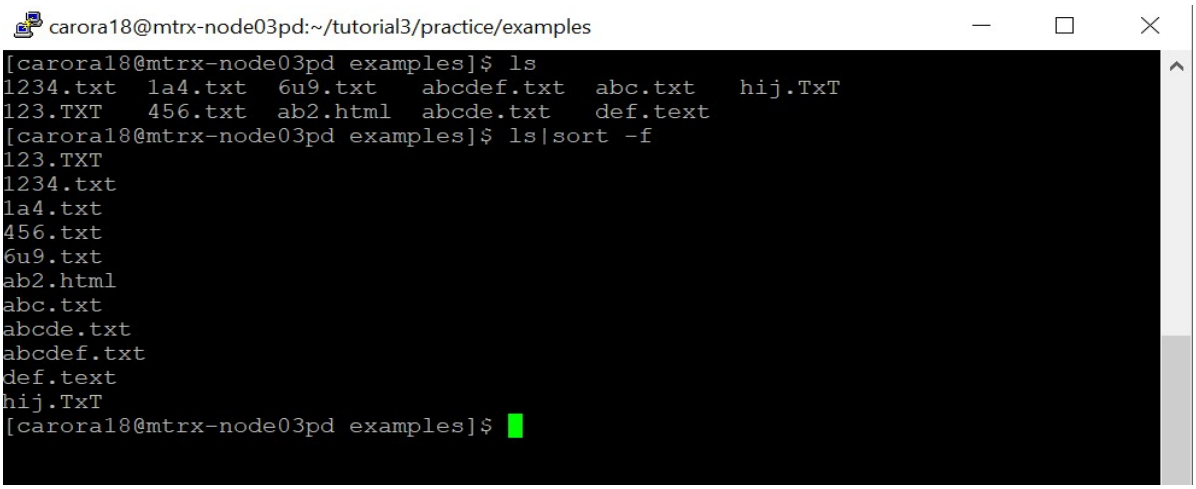
4. If you encounter errors, then make corrections (eg. **viewing directory contents**, **check for correct filename syntax**, **case sensitivity**, **missing files**, **files in the wrong location**, etc.)
   Ans: ls -l command is used to view detailed directory



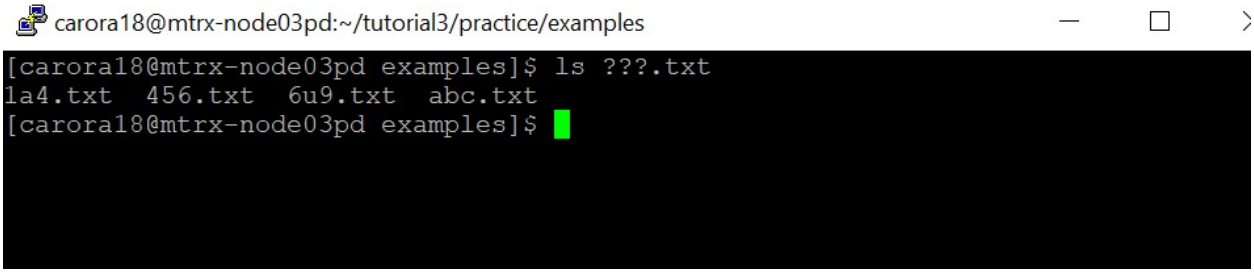
5. Issue the **ls** command to get a listing of files in your *examples* directory.
   The output should look identical to the diagram displayed below.
   You can refer to this listing to see all files so you can then predict the output from Linux commands that use filename expansion symbols.

```
ls
123.TXT  1234.txt  1a4.txt  456.txt  6u9.txt  ab2.html  abc.txt  abcde.txt  abcdef.txt  def.text  hij.TxT
```

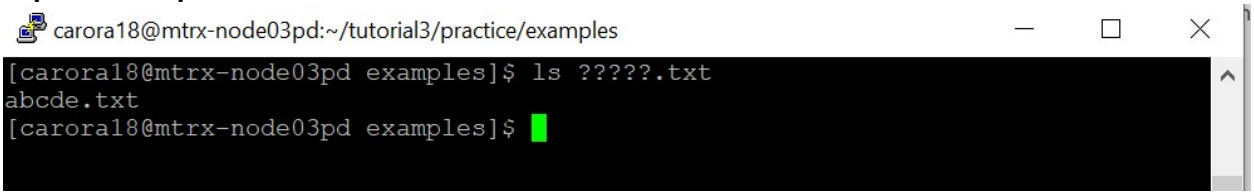
Ans:-



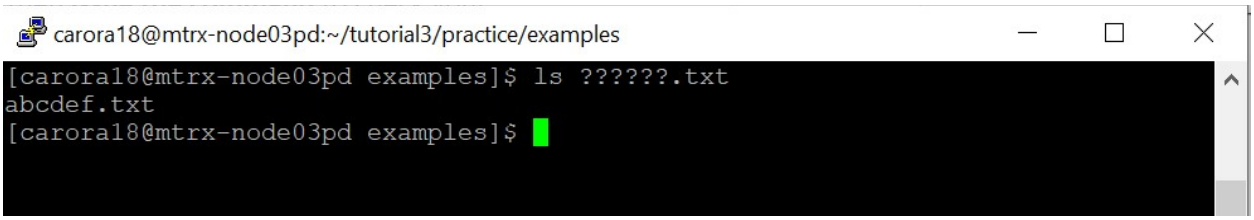
6. What do you think the output will be from the following Linux command?
   ls ???.txt
   Write down the expected output on paper, then **issue the command** to check your answer.
   Ans:- ? symbol define as one character. So, ??? defines the three characters in file. ls command it used to see the list of files in directory. ls ???.txt the result of this command is it shows the text files in directory which have three characters followed by .txt in their name.
   Expected output:- 1a4.txt 456.txt 6u9.txt abc.txt



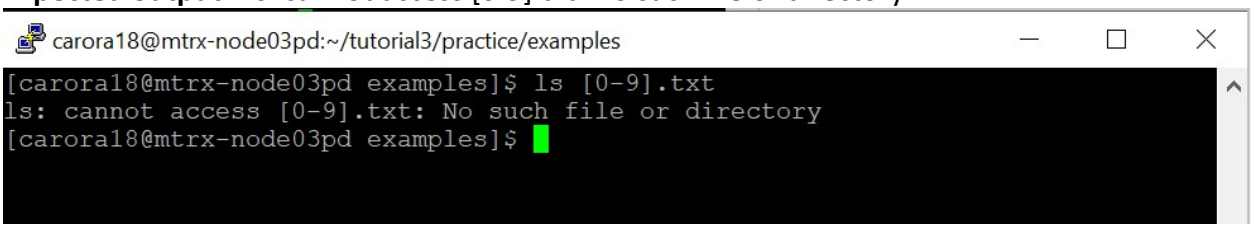
7. What do you think the output will be from the following Linux command?
   ls ?????.txt
   Write down the expected output on paper, then **issue the command** to check your answer.
   Ans: -? symbol define as one character. So, ????? defines the five characters in file. ls command it used to see the list of files in directory. ls ?????.txt the result of this command is it shows the text files in directory which have five characters followed by .txt in their name.
   Expected output:- abcde.txt



8. What do you think the output will be from the following Linux command?
   ls ??????.txt
   Write down the expected output on paper, then **issue the command** to check your answer.
   Ans: -? symbol define as one character. So, ?????? defines the six characters in file. ls command it used to see the list of files in directory. ls ??????.txt the result of this command is it shows the text files in directory which have six characters followed by .txt in their name.
   Expected output:- abcdef.txt

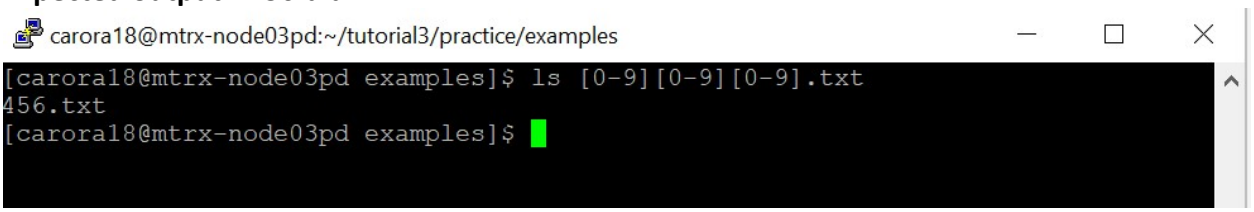


9. What do you think the output will be from the following Linux command?
   ls [0-9].txt
   Write down the expected output on paper, then **issue the command** to check your answer.br>Did the command work?
   What does this teach you about the character class [ ] symbol?
   Ans: - This command will list all the files in the current directory that have a single digit followed by .txt as their name.
   Expected output:- ls: cannot access [0-9].txt: No such file or directory



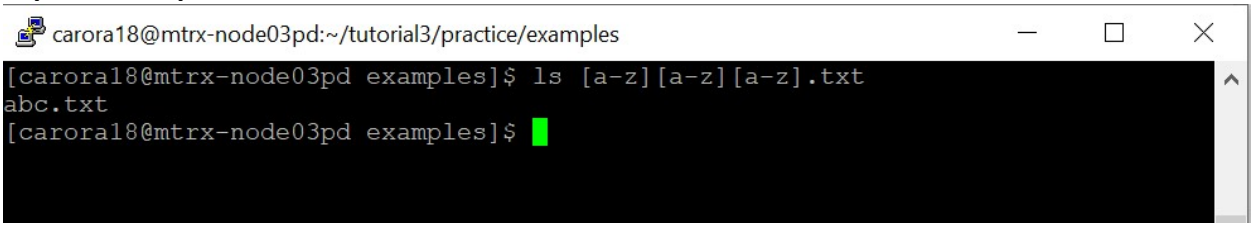
   This command did not worked.
   Square brackets ([ ]) to represent and match for the character enclosed within the square brackets.It represents ONLY ONE character - it's like a Question Mark (?) but with conditions or restrictions.

10. What do you think the output will be from the following Linux command?
    ls [0-9][0-9][0-9].txt
    Write down the expected output on paper, then **issue the command** to check your answer.
    Ans: - This command will list all the files in the current directory that have a three digits followed by .txt as their name.
    Expected output:- 456.txt

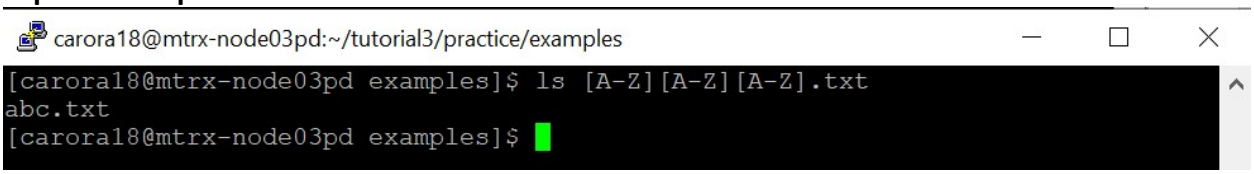


11. What do you think the output will be from the following Linux command?
    ls [a-z][a-z][a-z].txt
    Write down the expected output on paper, then **issue the command** to check your answer.
    Ans: - This command will list all the files in the current directory that have a three charaters(letters) in lower case from a to z followed by .txt as their name.

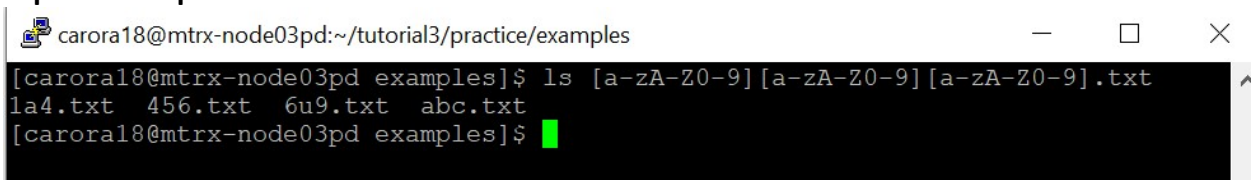
Expected output:- abc.txt



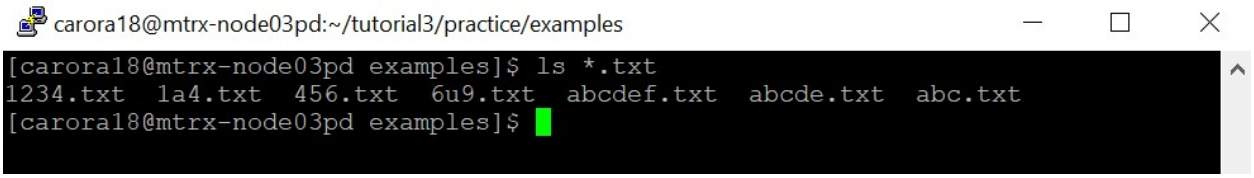
12. What do you think the output will be from the following Linux command (using character class with UPPERCASE letters)?:
    ls [A-Z][A-Z][A-Z].txt
    Write down the expected output on paper, then **issue the command** to check your answer.
    Ans: - This command will list all the files in the current directory that have a three charaters(letters) from A to Z followed by .txt as their name. there is no any difference to type upper or lower character in [] braces.
    Expected output:- abc.txt



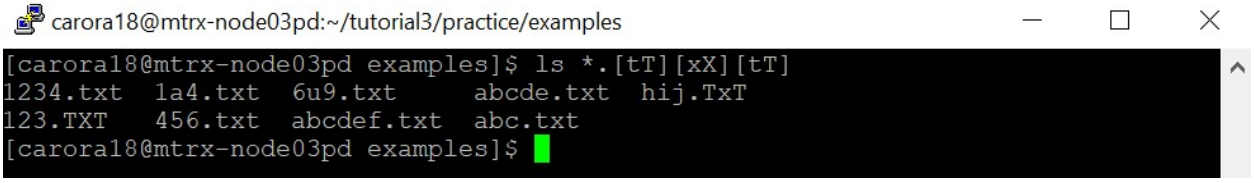
13. What do you think the output will be from the following Linux command (using character class using alpha-numeric characters)?
    ls [a-zA-Z0-9][a-zA-Z0-9][a-zA-Z0-9].txt
    Write down the expected output on paper, then **issue the command** to check your answer.
    Ans: - This command will list all the files in the current directory that have a three-character name consisting of any combination of letters and digits, followed by .txt as their extension.
    Expected output:- 1a4.txt 456.txt 6u9.txt abc.txt



14. What do you think the output will be from the following Linux command?
    ls *.txt
    Write down the expected output on paper, then **issue the command** to check your answer. Did ALL text files get listed? Why not?
    Ans: - This command will list all the files in the current directory that have .txt as their extension.
    Expected output:- 1234.txt 1a4.txt 456.txt 6u9.txt abcdef.txt abcde.txt abc.txt

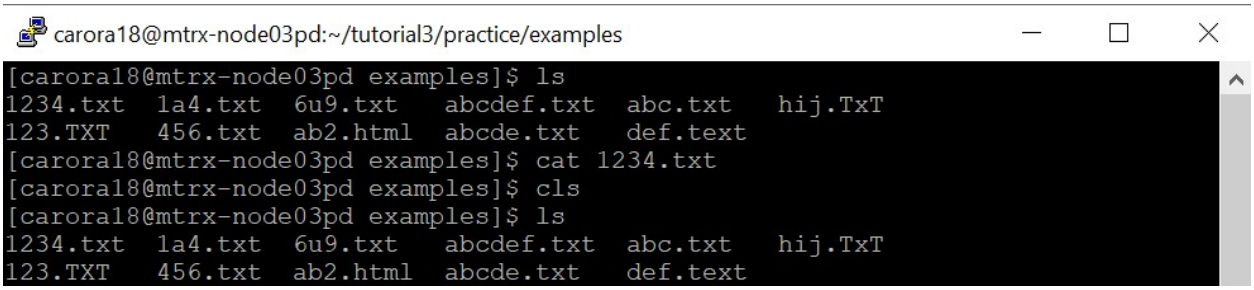


15. What do you think the output will be from the following Linux command?
    ls *.[tT][xX][tT]
    Write down the expected output on paper, then **issue the command** to check your answer. Did ALL text files get listed this time? If so, why?
    Ans: - This command will list all the files in the current directory that have `.txt` or `.TXT` as their extension, regardless of the case.
    Expected output:- 1234.txt 1a4.txt 6u9.txt abcde.txt hij.TxT 123.TXT 456.txt abcdef.txt abc.txt

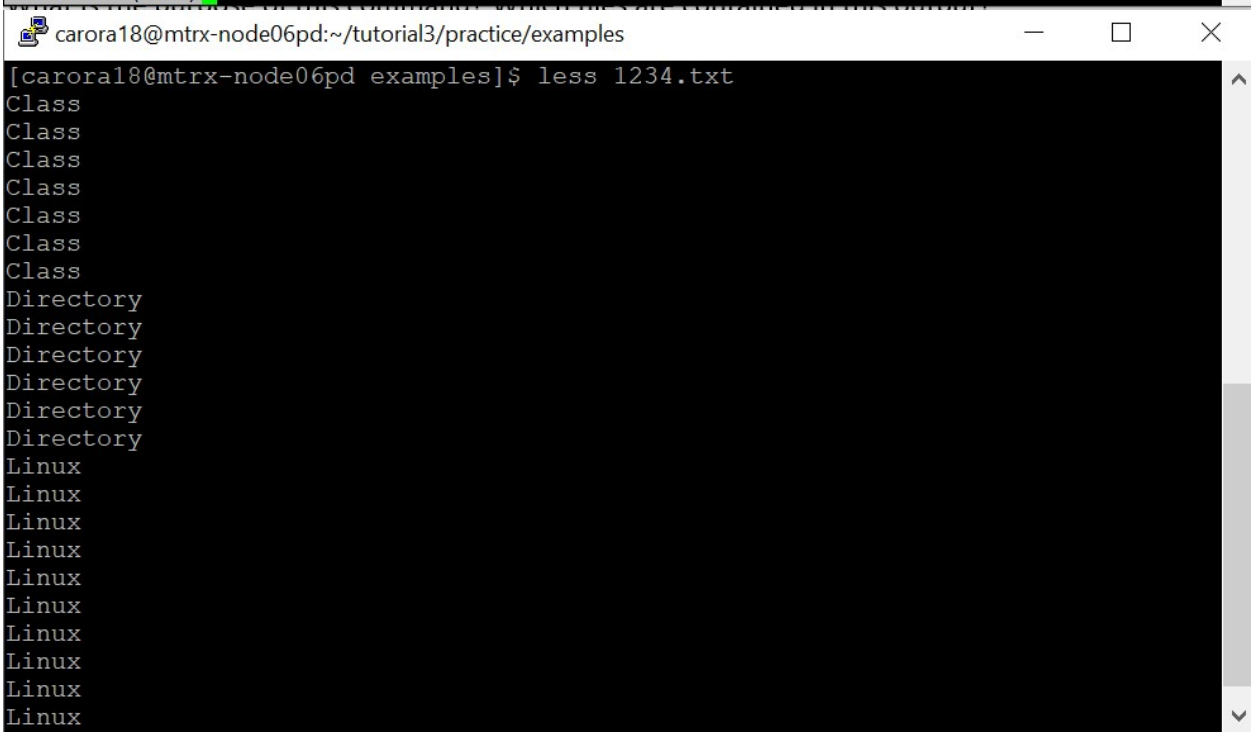


    Yes all the files listed this time.

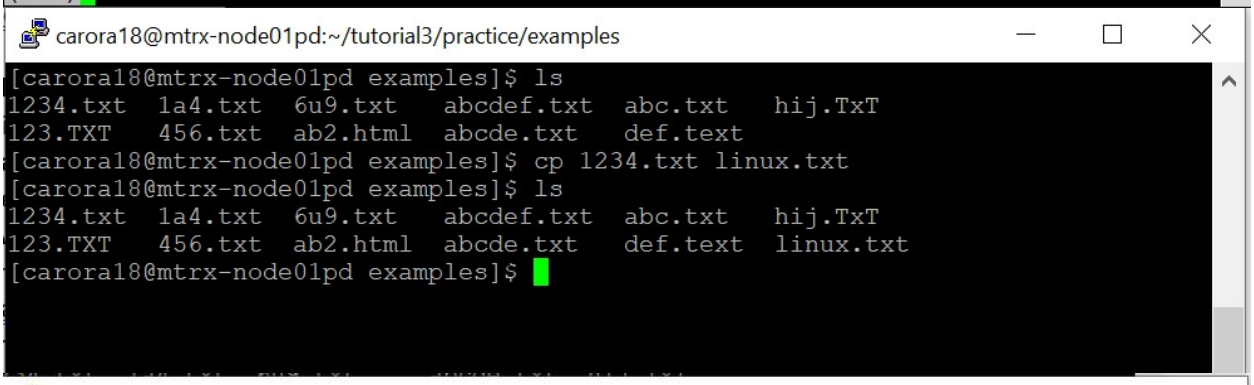
16. **NOTE:** We have just been using filename expansion symbols just with the ls command. Filename expansion symbols can be used for ANY Linux file management command (e.g. **cat**, **more**, **less**, **cp**, **mv**, **rm**, **ls**, etc.).
    Let's get some practice issuing these other Linux file management commands.
    Ans: - Filename expansion symbols are useful for performing operations on multiple files that match a certain pattern. Some are below:-
    cat: - cat command is used to view the content of the file.
    more: - more command is used to display in forward page by page.
    less: - less command is used to display in backward page by page.
    cp: - cp command is used to copy the file as well as directory.
    mv: - mv command is used to move the file as well as directory.
    rm: - rm command is used to remove/delete the file as well as directory.
    ls: - ls command is used to view the content in the file.

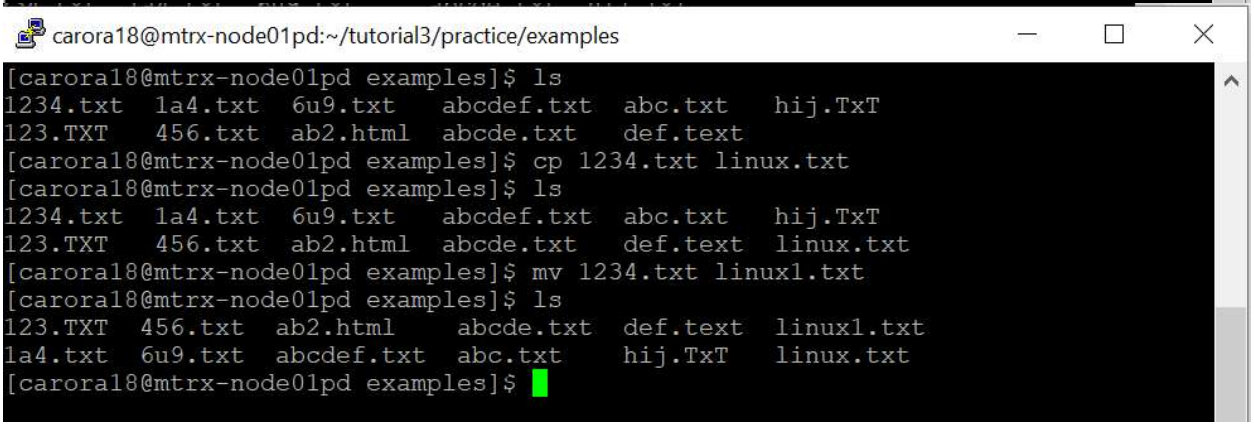



carora18@mtrx-node03pd:~/tutorial3/practice/examples
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
Class
Class
Class
Class
Class
Class
Class
Class
Directory
Directory
Directory
--More--(62%)


carora18@mtrx-node06pd:~/tutorial3/practice/examples
[carora18@mtrx-node06pd examples]$ less 1234.txt
Class
Class
Class
Class
Class
Class
Class
Directory
Directory
Directory
Directory
Directory
Directory
Linux
Linux
Linux
Linux
Linux
Linux
Linux
Linux
Linux
Linux


carora18@mtrx-node06pd:~/tutorial3/practice/examples
Class
Class
Class
Class
Class
Class
Class
Directory
Directory
Directory
Directory
Directory
Directory
Linux
Linux
Linux
Linux
Linux
Linux
Linux
Linux
Linux
Linux
(END)


carora18@mtrx-node01pd:~/tutorial3/practice/examples
[carora18@mtrx-node01pd examples]$ ls
1234.txt 1a4.txt 6u9.txt abcdef.txt abc.txt hij.TxT
123.TXT 456.txt ab2.html abcde.txt def.text
[carora18@mtrx-node01pd examples]$ cp 1234.txt linux.txt
[carora18@mtrx-node01pd examples]$ ls
1234.txt 1a4.txt 6u9.txt abcdef.txt abc.txt hij.TxT
123.TXT 456.txt ab2.html abcde.txt def.text linux.txt
[carora18@mtrx-node01pd examples]$

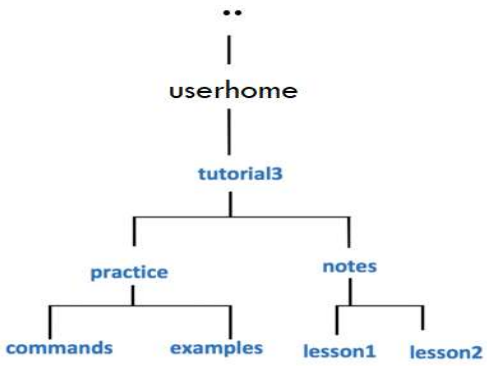

carora18@mtrx-node01pd:~/tutorial3/practice/examples
[carora18@mtrx-node01pd examples]$ ls
1234.txt 1a4.txt 6u9.txt abcdef.txt abc.txt hij.TxT
123.TXT 456.txt ab2.html abcde.txt def.text
[carora18@mtrx-node01pd examples]$ cp 1234.txt linux.txt
[carora18@mtrx-node01pd examples]$ ls
1234.txt 1a4.txt 6u9.txt abcdef.txt abc.txt hij.TxT
123.TXT 456.txt ab2.html abcde.txt def.text linux.txt
[carora18@mtrx-node01pd examples]$ mv 1234.txt linux1.txt
[carora18@mtrx-node01pd examples]$ ls
123.TXT 456.txt ab2.html abcde.txt def.text linux1.txt
1a4.txt 6u9.txt abcdef.txt abc.txt hij.TxT linux.txt
[carora18@mtrx-node01pd examples]$

```
carora18@mtrx-node01pd:~/tutorial3/practice/examples
[carora18@mtrx-node01pd examples]$ ls
1234.txt  1a4.txt  6u9.txt   abcdef.txt  abc.txt   hij.TxT
123.TXT   456.txt  ab2.html  abcde.txt   def.text
[carora18@mtrx-node01pd examples]$ cp 1234.txt linux.txt
[carora18@mtrx-node01pd examples]$ ls
1234.txt  1a4.txt  6u9.txt   abcdef.txt  abc.txt   hij.TxT
123.TXT   456.txt  ab2.html  abcde.txt   def.text  linux.txt
[carora18@mtrx-node01pd examples]$ mv 1234.txt linux1.txt
[carora18@mtrx-node01pd examples]$ ls
123.TXT  456.txt  ab2.html    abcde.txt  def.text  linux1.txt
1a4.txt  6u9.txt  abcdef.txt  abc.txt    hij.TxT   linux.txt
[carora18@mtrx-node01pd examples]$ rm linux.txt
[carora18@mtrx-node01pd examples]$ ls
123.TXT  456.txt  ab2.html    abcde.txt  def.text  linux1.txt
1a4.txt  6u9.txt  abcdef.txt  abc.txt    hij.TxT
[carora18@mtrx-node01pd examples]$
```

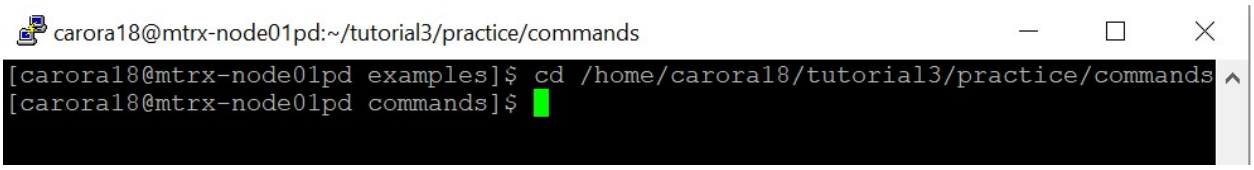
17. Issue the following Linux command: **file *.[tT][xX][tT]**
    What is the purpose of this command? Which files are contained in this output?
    Ans:- This command display the file type of all the files in the current directory that have .txt or .TXT as their extension, regardless of the case.

```
carora18@mtrx-node01pd:~/tutorial3/practice/examples
[carora18@mtrx-node01pd examples]$ file *.[tT][xX][tT]
123.TXT:    empty
1a4.txt:    empty
456.txt:    empty
6u9.txt:    empty
abcdef.txt: empty
abcde.txt:  empty
abc.txt:    empty
hij.TxT:    empty
linux1.txt: ASCII text
[carora18@mtrx-node01pd examples]$
```

18. Change to the **commands** directory using an **absolute** pathname
    (use the diagram on right-side for reference).

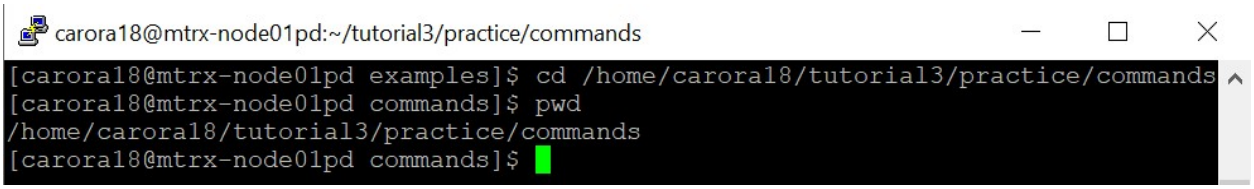


Ans:- cd /userhome/tutorial3/practice/commands



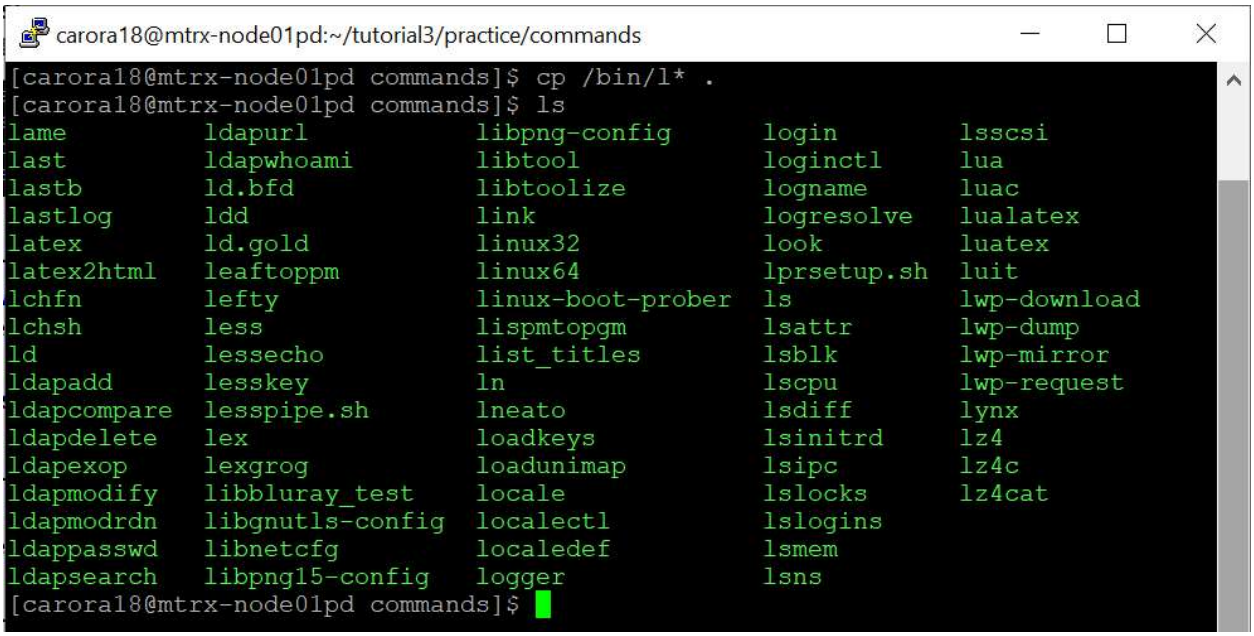
19. Issue a Linux command to confirm that you are now in the **commands** directory.

Ans: -



20. Issue the following Linux command (lowercase "l" NOT the number "1"):
cp /bin/l* .
View the contents of the contents directory. What did this command do?

Ans: -

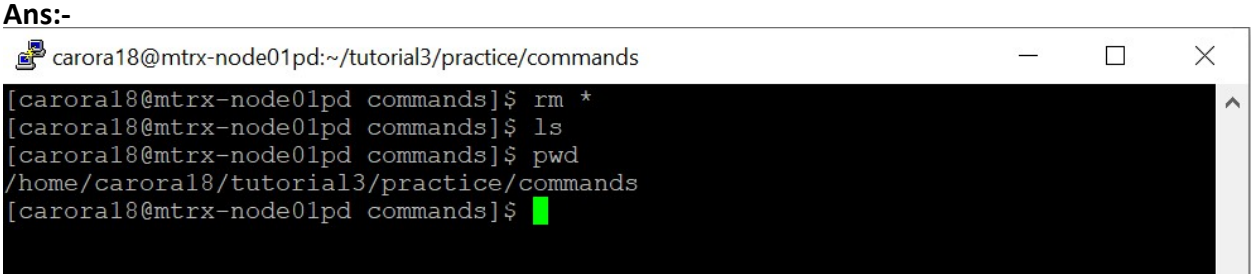


This command copy all the files in the /bin directory that start with the letter l to the current working directory (.).

21. Issue the following Linux command: **rm ***

View the contents of the contents directory. What did this command do?

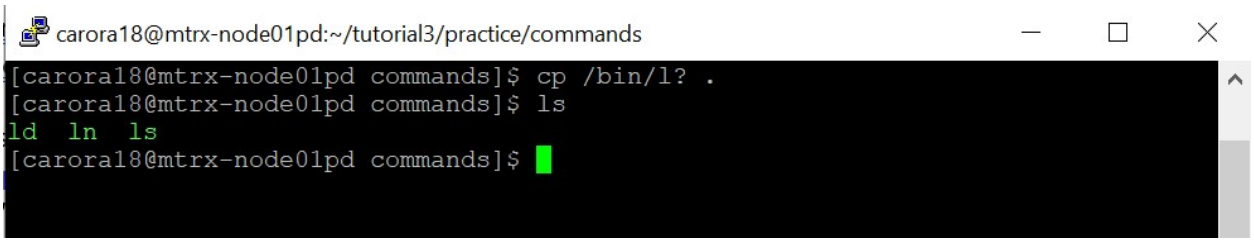
Ans:-



rm command is used to remove/delete the files. rm* command deleted all the files from commands

22. Issue the following Linux command (lowercase "l" NOT the number "1"):
    cp /bin/l? .
    View the contents of the contents directory. What did this command do?

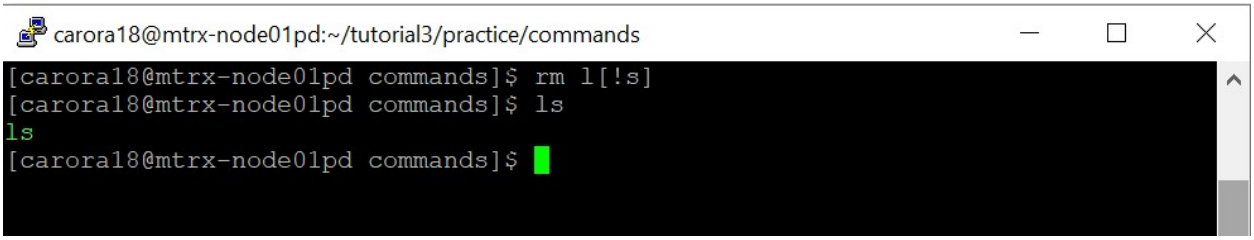
Ans: -



This command copies all the files in the /bin directory whose names start with l and have one more character after that to the current working directory (.).

23. Issue the following Linux command: **rm l[!s]**
    View the contents of the contents directory. What did this command do?

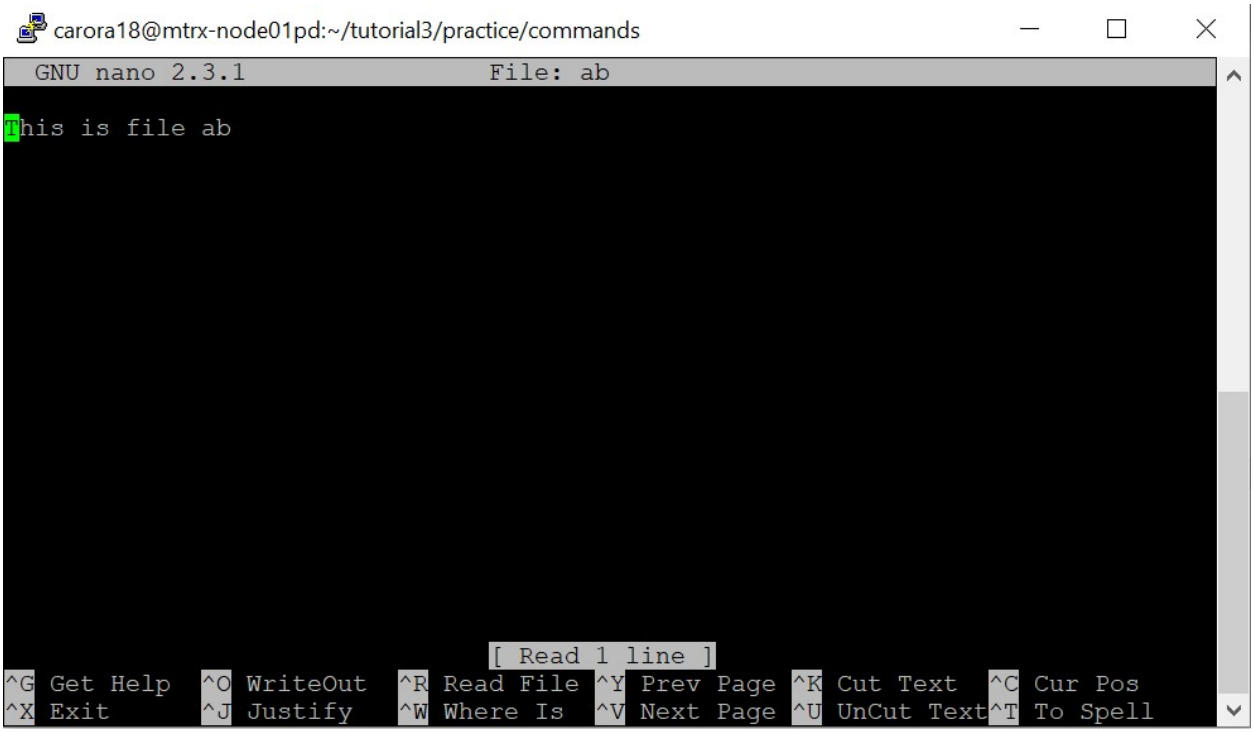
Ans: -

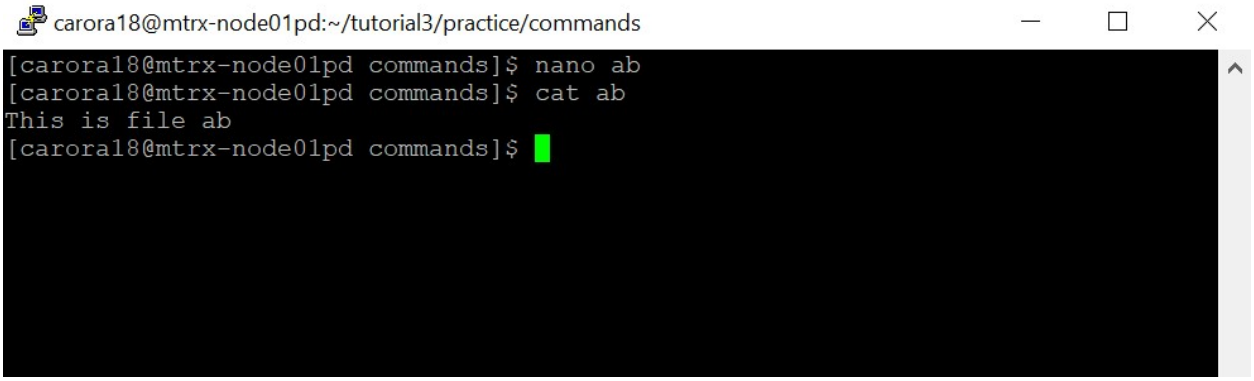


This command removes all the files in the current directory whose names start with l and do not end with s.

24. Use a text editor (nano or vi) to create the file called **ab** in the **commands** directory that contains the line of text below,
    and then save editing changes to this file:
    This is file ab

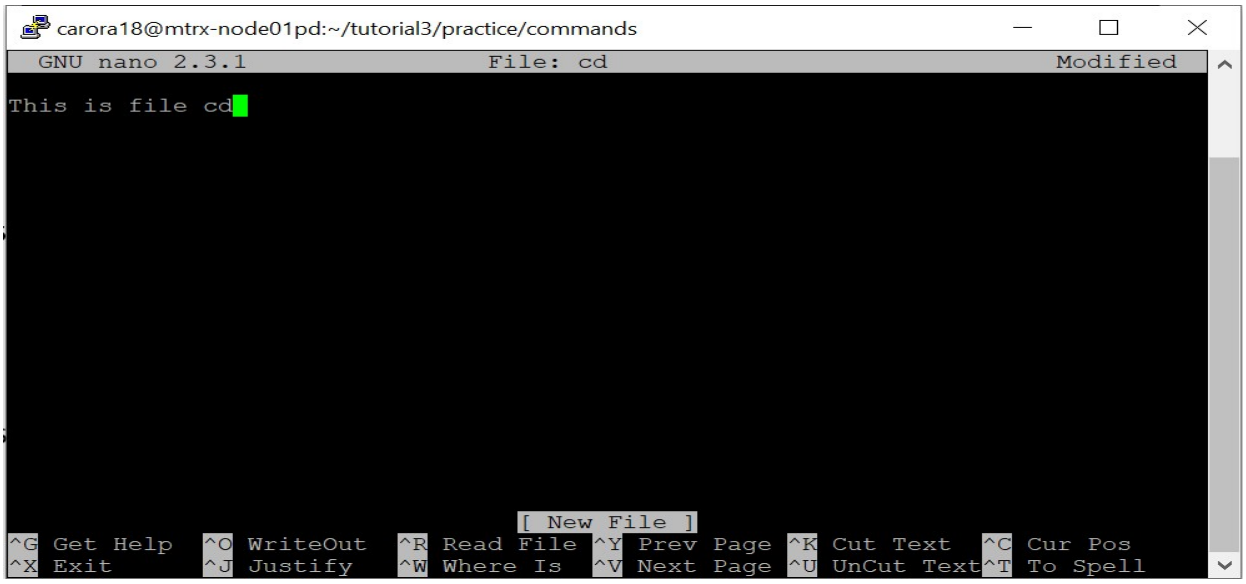
Ans: -

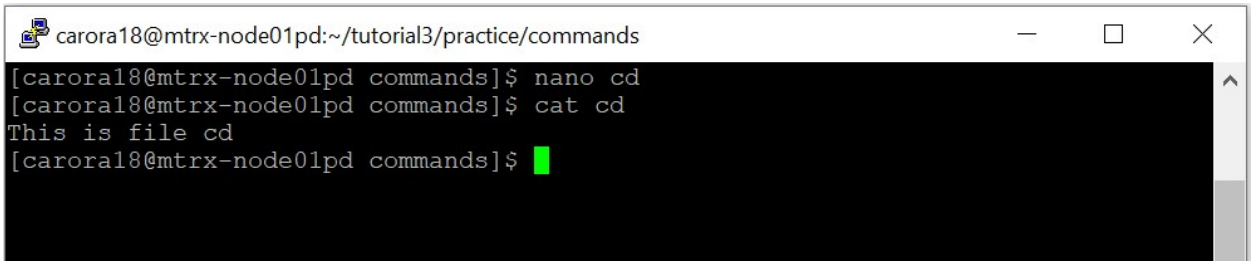




25. Use a text editor (nano or vi) to create the file called **cd** in the **commands** directory that contains the line of text below,
and then save editing changes to this file:
This is file cd

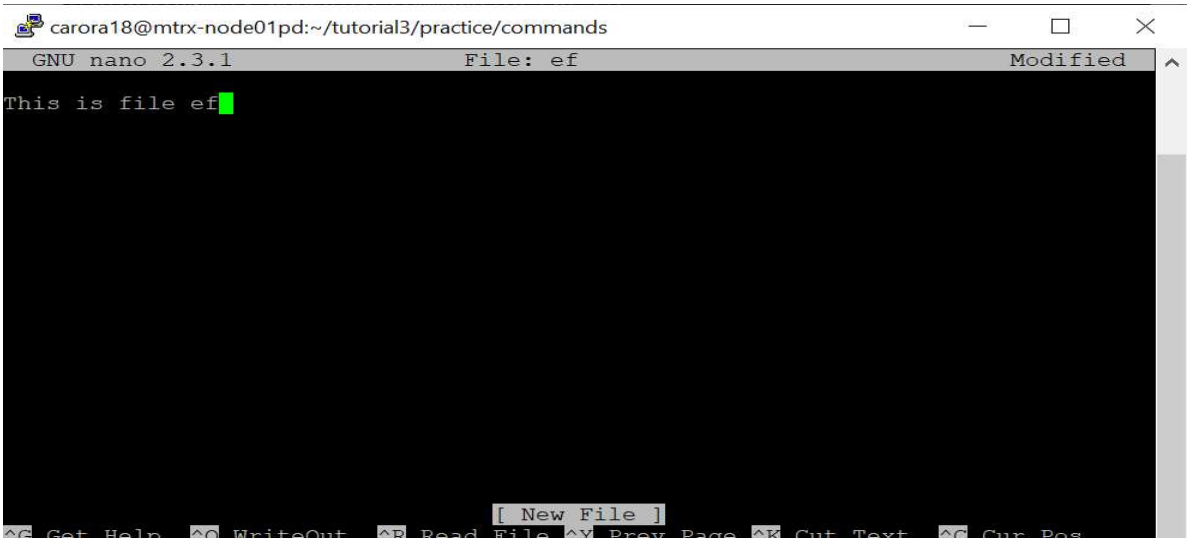
Ans: -

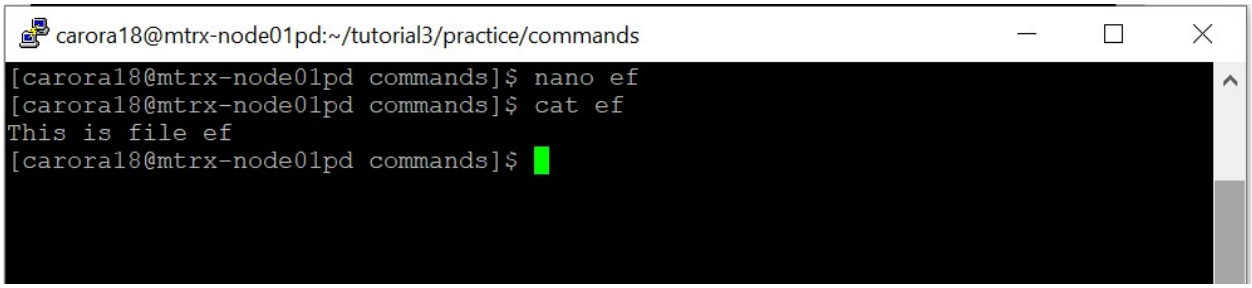




26. Use a text editor (nano or vi) to create the file called **ef** in the **commands** directory that contains the line of text below,
and then save editing changes to this file:
This is file ef

Ans: -



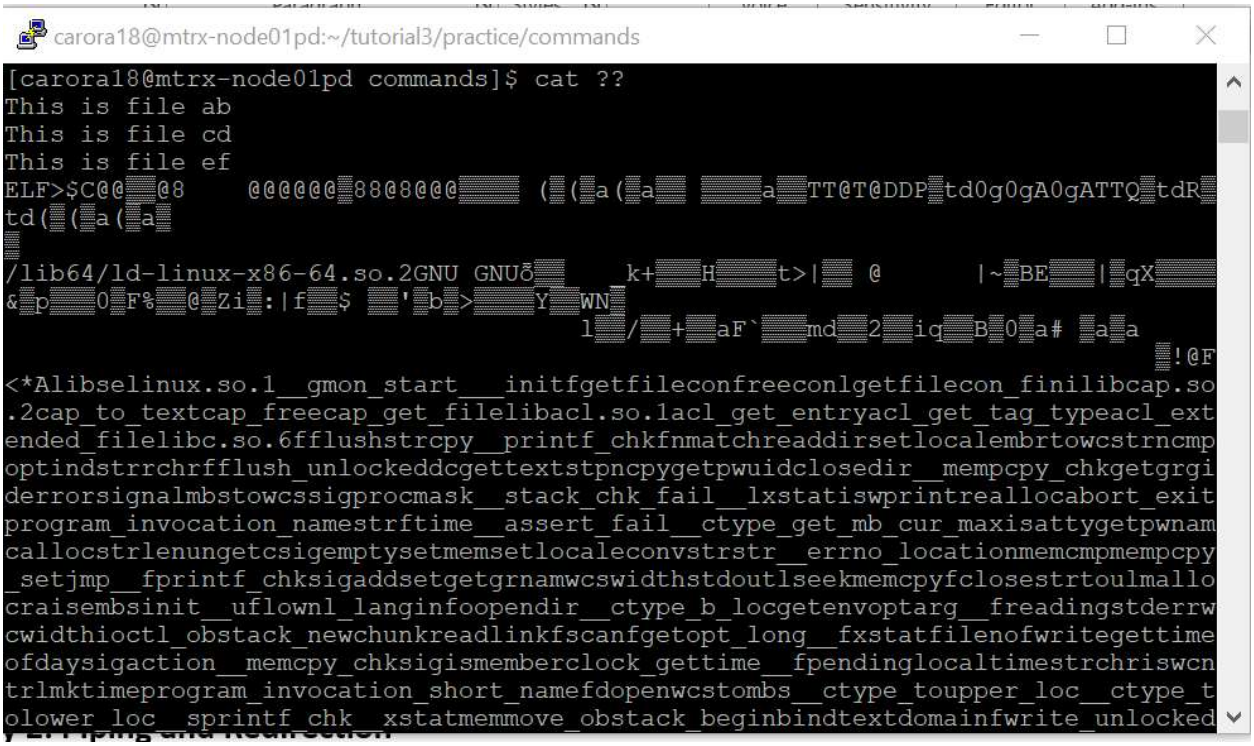


27. Issue the following Linux command: **cat ??**

    View the contents of the contents directory. What did this command do? Why does the output look strange?

    NOTE: Press the keys **ctrl-c** to return to the shell prompt.

    Ans: -



This command displays the contents of all the files in the current directory whose names have exactly two characters. The ?? part is a pattern that matches any two characters. The output of the cat ?? command might look strange because it concatenates the contents of all the matching files without any separation or indication of which file they belong to.

28. Issue the following Linux command: **cat [!l][!s]**

    View the contents of the contents directory. What did this command do? Does the output look better? If so, why?

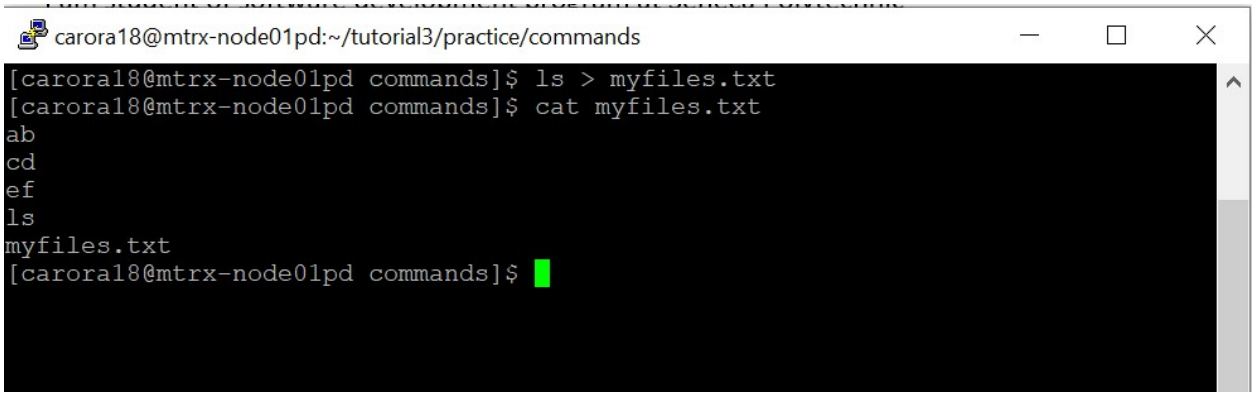
    Ans: -

    This command displays the contents of all the files in the current directory whose names do not start with l and do not end with s. The [!l] part is a pattern that matches any single character except l, and the [!s] part is a pattern that matches any single character except s. Yes, the output look better because the command excludes the files whose names start with l and end with s, which might have different formats or contents than the other files.

Activity 2: Piping and Redirection

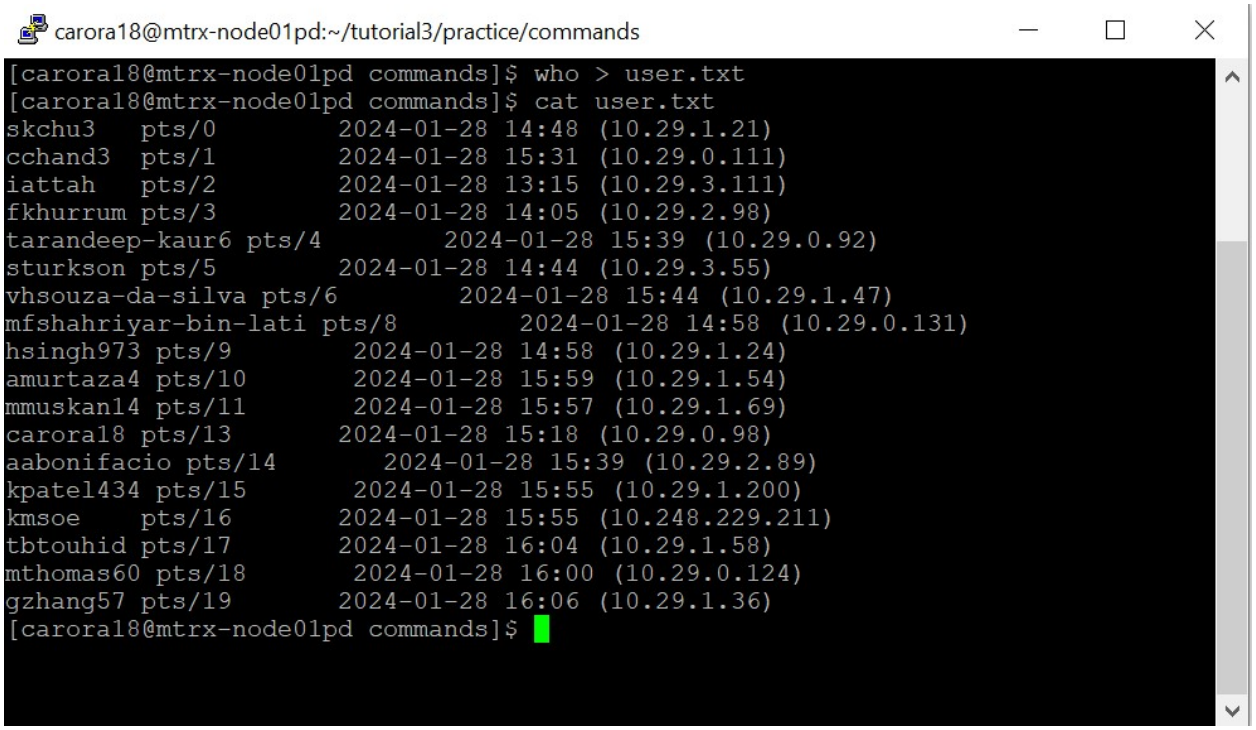
1. Write command to transfer the output of ls command to a file myfiles.txt

    Ans: -



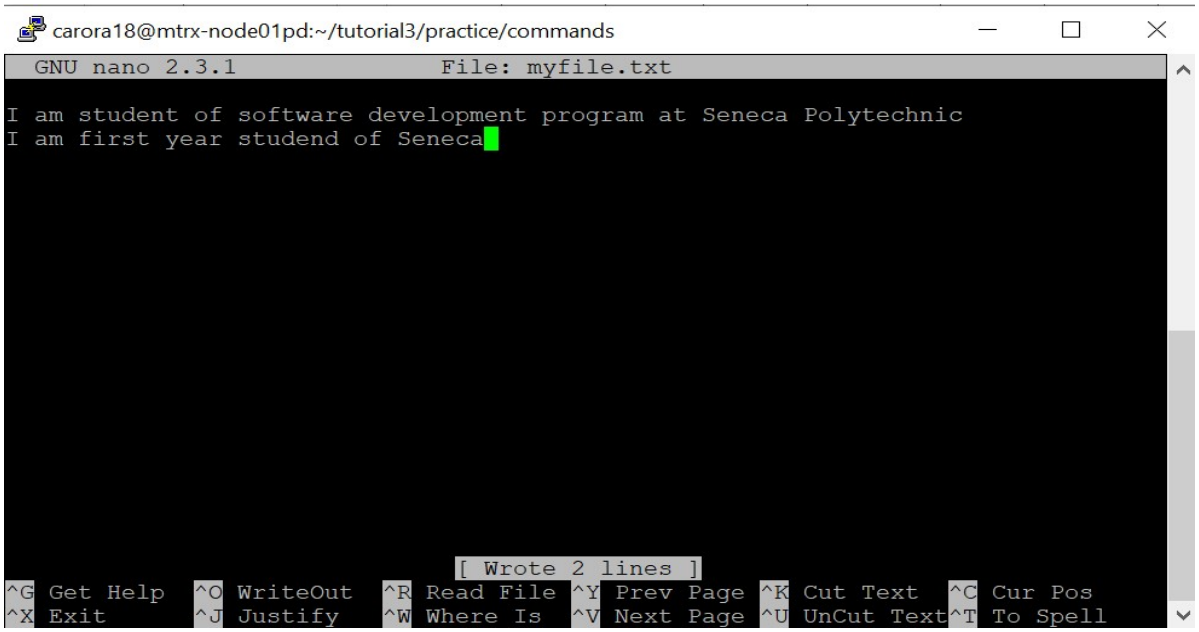
2. Write command to transfer the output of who command to a file *user.txt*

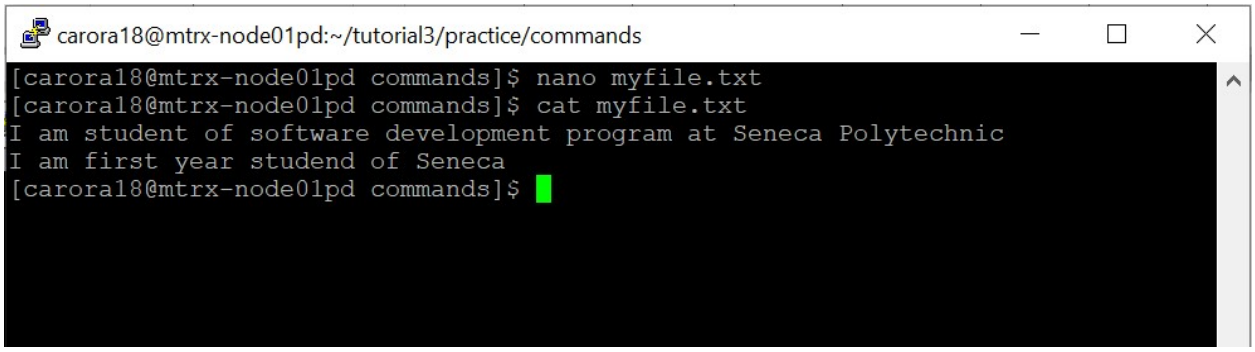
Ans: -



3. Using nano editor to write following text in a file *myfile.txt*
   I am student of software development program at Seneca Polytechnic
   I am first year student of Seneca

Ans: -





4. Write command to search word Seneca in file myfile.txt

   Ans: -