We will use the below data file for this quiz.

**"datafile"**

----------------------------------------

Name:DateOfBirth:Salary:Hours-worked

joey:10-05-1994:65000:40

peter:04-13-1990:50000:40

sy:02-22-1999:1000000:20

-----------------------------------------

1. **file=datafile ;echo $file**

3. What do you expect when you execute the following command?

**$ cut -d: -f3 $file**

Salary

65000

50000

1000000

4. Write a cut command to get the name and salary from datafile?

**$ cut -d: -f 1,3 $file**

5. What happens when you execute this command?

**$ cut -d: -f3 < $file**

Salary

65000

50000

1000000

6.  Get the first five characters of each line in the datafile

**$ cut -c '10-15'  $file**

**$ls**

**Using the 'tr' command**

      [ tr - translate or delete characters ]

**SYNOPSIS**

        [ **tr** [OPTION]... SET1 [SET2] ]

  Create this names.txt  file with the names listed.

--------------

joe richards

mac arther

joe richard

lynn  nguyen

fenj Leu

----------------

7. Write a command using 'tr' to uppercase all names in the file.

**$ cat $file | tr 'a-z'  'A-Z'**

8. Replace all lower case characters in the file with an 'X'.

**$ cat $file | tr '[a-z]' 'X'**

9.  Write a sed or other command to mask the salary in the data file so that the output of each line looks like this below:

**joey:10-05-1994:\*\*\*\*\*:40**

**$ sed -E 's/:[0-9][0-9]\*:/:\*\*\*\*\*:/g' $file**

Or $ sed 's/\:[0-9][0-9]\*\:/\:\*\*\*\*\*\:/g' $file

**$file is datafile**

10. Write a command to count the number of lines in the datafile.

 $wc -l **$file**

 $**sed -n '$=' $file**

11. Write a command to count the numbers of characters in the datafile.

$wc -m **$file**

12. When you run the below command on datafile, the output seen is listed below. For full credit, explain what is happening in detail as to why you got the output below.

**$ sed '1h;2d;3H;4g' datafile**

name:dob:salary:hours-worked

peter:04-13-1990:50000:40

name:dob:salary:hours-worked

peter:04-13-1990:50000:40

   1step:  print line 1and copy line 1,

2step: delete line2

3step: print line 3 and append line3(add after line1)

4step: paste and print line 1 and 3 (overwriting 4)

* Look at the original datafile and describe what happened to line number 4 of the datafile?

It changed to line 3and4

13.  Write a sed command to produce the output below from the datafile. The command will look very similar to number 12.

name:dob:salary:hours-worked

peter:04-13-1990:50000:40

sy:02-22-1999:1000000:20

name:dob:salary:hours-worked

**$ sed '1h;2d;4G' datafile**

14.  Write a sed command to split up the datafile so that the first line goes into a file called 'df1', the second and third line will go into a file called 'df2' and the fourth line will go into a file called 'df3'.

$ sed -e '1 w df1' -e '2,3 w df2' -e '4 w df3' datafile

15. Write a sed command to match lines in a file that does not begin with a space or a tab from a file called xfile. To get a space, just hit the space bar. To get a Tab, you should use the Contrl+letter v, then hit the Tab key.

**$ sed -n '/^[ ^Space ctrl+v Tab]/p'  xfile**

16. Write a sed command to match lines that begins with a space or a tab

**$ sed -n '/^[Space ctrl+v Tab]/p'  xfile**

17. Write a sed command to delete lines 1 to 3 in datafile

**$ sed   '1,3d'   $file**

18. Write a sed command to delete lines 2 to the end of the datafile.

**$ sed   '2,$d'  $file**

19. Given the two files: filea and fileb

 filea                                                      fileb

I am one                                       I am one

I am one                                       I am one

I AM LINE TWO                     I AM LINE TWO

I am line two                             I am line two

I am line 3 in filea                   I am line 3 in fileb

* Write a command to show the differences between the two files.

**$ cat filea fileb | sort | uniq -u**

* Write a command to show only unique lines in the each file.
  + **$ uniq filea; uniq fileb**
* What does the -i option do ?

**$ uniq -i filea**

 ignore differences in case when comparing

* What does the -c option do?

**$ uniq -i -c  filea**

prefix lines by the number of occurrences

20.  Write a find command to find all regular files under your home directory belonging to your LONGNAME that were created WITHIN the last 3 days that are between the size of 4 to 5 Megabytes and when you find them, you should tar them up.

$find $LONGNAME -type f -a -mtime -3 -a -size +4M -a -size -5M -exec tar cvf mfiles.tar {} \;

21. Write a find command and use it in conjunction with xargs to find then name of any Regular file under your home directory that has the word 'xyz' inside of the file.

$find $HOME -type f | xargs grep **-RnH** “xyz”

22. Write a command to find which file under the $HOME directory have lines that are 80 characters or more long.

**$ egrep -Rl "^.{80,}$"  $HOME**

23. Let's see the lines that actually matches

**$ egrep -RnH "^.{80,}$" $HOME**

24.  Write a recursive grep command to search all files in your home that has the word 'homework'

 $ grep -Rn "homework" $HOME

25.  Write a find to find all files with inode number 12345 and move them to the /tmp/storage directory.

 $find $HOME -inum 12345 -exec cp {} /tmp/storage \;