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

**datafile**

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

$ cut -d: -f3 $file

**Salary**

**65000**

**50000**

**1000000**

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

**$ cut -d: -f1,3 $file**

4. What happens when you execute this command?

     $ cut -d: -f3 < $file

**The following output is generated:**

**Salary**

**65000**

**50000**

**1000000**

**This means that the third field of the $file is cut out and sent to the console**

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

**$ cut -c '1-5' $file**

 6. 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 names.txt | tr 'a-z' 'A-Z'**

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

**$ cat names.txt | 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 's/^\([a-zA-Z][a-zA-Z]\*\)\(:\)\([0-9]\{2\}\)\(-\)\([0-9]\{2\}\)\(-\)\([0-9]\{4\}\)\(:\)\([0-9][0-9]\*\)/\1\2\3\4\5\6\7\8\\*\*\*\*\*/' datafile**

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

**$ wc -l $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

     line one of datafile is copied to hold space, line two is deleted, line three is appended to hold space, then everything in the hold space is copped to pattern space.

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

 There is no command to copy or append or delete or print line number 4 of the data file.

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

**$ sed ‘1h;2d;3h;4g’ datafile**

name:dob:salary:hours-worked

peter:04-13-1990:50000:40

sy:02-22-1999:1000000:20

name:dob:salary:hours-worked

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 -ie ‘s/^Space/p; s/^ctr+v Tab/p’ datafile

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.

  $ diff “filea” “fileb”

* Write a command to show only unique lines in each file.

$ uniq filea; uniq fileb

* What does the -i option do ?

              $ uniq -i filea

* What does the -c option do?

               $ uniq -i -c  filea

20.  Write a find command to find all regular files under your home directory belonging to your LOGNAME 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 $HOME/ -type f -user ‘cs45aa15’ -mtime -3 -size +4M -o -size -5M -exec tar cvf userfiles.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 -il ‘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 -R ‘homework’ $HOME

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

$ find / -inum 12345 -exce mv {} /tmp/storage \;