

CIS-45 FALL 2020

Quiz-6

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

**Print the 3rd field in datafile separated by colon delimiters**

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

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

5. What happens when you execute this command?

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

**Print the 3rd field with input from datafile separated by colon delimiters**

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

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

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'
```

```
JOE RICHARDS
MAC ARTER
JOE RICHARD
LYNN NGUYEN
FENJ LEU
```

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

```
$ cat names.txt | tr 'a-z' 'X'
```

```
XXX XXXXXXXX
XXX XXXXXX
XXX XXXXXXXX
XXXX XXXXXXX
XXXX XXX
```

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

```
$ sed 's/^\([[:alpha:]]\([[:alpha:]]*\.[[:digit:]]\([[:digit:]]*\)\([[:digit:]]\([[:digit:]]*\)\$)\1\1\1\1\1\1\3/' datafile
```

```
$ sed 's/^\(.*:.*:\)\([0-9]*\)\(.*\)$/\1\*\*\*\*\3/' datafile
```

```
Name:DateOfBirth:Salary:Hours-worked
joey:10-05-1994:*****:40
peter:04-13-1990:*****:40
sy:02-22-1999:*****:20
```

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

```
$ grep -c " datafile
```

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

```
$ sed 's/[[:alnum:]][[:punct:]]/&\n/g' datafile | grep -c '[[:alnum:]][[:punct:]]'
```

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
```

1. Line1 is brought to pattern buffer, printed to output by default and brought to holding buffer.
2. Line2 is brought to pattern buffer then deleted.
3. Line3 is brought to pattern buffer, printed to output by default and appended to the holding buffer.
4. Line4 is brought to pattern buffer, where it is replaced (g) by Line1 and Line3 being brought back to the pattern buffer. Line1 and Line3 are printed to output from the pattern buffer.

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 '1w df1' -e '2,3w df2' -e '4w 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 Ctrl+letter v, then hit the Tab key.

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

```
$ sed -n '/^[^      ]/p' space.txt
```

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

```
$ sed -n '/^[ \t]/p' space.txt
```

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

```
$ sed '1,3d' datafile
```

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

```
$ sed '2,$d' datafile
```

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

=> outputs filea with only unique lines ignoring case

```
filea
I am one
I AM LINE TWO
I am line 3 in filea
```

- What does the -c option do? **count -> prefixes lines by occurrence**  
\$ uniq -i -c filea

=> outputs filea with only unique lines that ignore case and are prefixed by line occurrence

```
1 filea
2 I am one
2 I AM LINE TWO
1 I am line 3 in 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 cs45aa08 -ctime -3 -size +4 -size -5 | tar cvf files.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 -l '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
```

```
$ find $HOME -type f -size +80c
```

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'

```
$ find ~ -type f | xargs grep -i 'homework'
```

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

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