Question 1: The file ‘words.txt‘ contains a single token per line. There are lots of duplicate tokens in this file i.e. the same token is present in multiple lines. Write a command to create a file named ‘vocab.txt’ which only contains unique tokens present in the ‘words.txt’ file, where every line contains only one token.

Correct solutions:

a. cat words.txt | sort | uniq > vocab.txt
b. cat words.txt | sort -u > vocab.txt
c. grep -o -E '\w+' words.txt | sort -u > vocab.txt
d. grep -o -E '\w+' words.txt | sort | uniq > vocab.txt
e. There can be more such one liners. The vocab file should contain around 28k to 29k words.

Question 2: How will you find all the files whose names have the prefix ‘main’ present in the directory ‘mystery’. The files may be present in any of its subdirectories.

Correct solutions:

a. find . -name "main*"
b. find mystery/ -name "main*"
c. There can be more such one liners.

Expected answer:

Six files or six file paths:

mystery/m/a/2/0/1/main.o
mystery/c/s/2/0/2/main.math
mystery/c/s/2/0/1/main.data
mystery/c/s/2/2/1/main.digital
mystery/c/s/2/1/0/main.lab
mystery/c/s/2/4/2/main.c

or similar to

main.o
main.math
main.data
main.digital
main.lab
main.c

Question 3: How will you find all the files present in the cricket_commentary_dataset directory which contain the words “kohli” and “chahal” in them. Here the words are present inside the file, not in the file names. You have to only list file names or file paths not the matched pattern.

Correct solutions:

a. grep -E "kohli" cricket_commentary_dataset/* | grep "chahal" | cut -d ':' -f 1
b. grep -E "chahal" cricket_commentary_dataset/* | grep "kohli" | cut -d ':' -f 1
c. grep -E "kohli.*chahal|chahal.*kohli" cricket_commentary_dataset/* | cut -d ':' -f 1

d. Any other command that gives the following output:

```
cricket_commentary_dataset/iw
cricket_commentary_dataset/lf
cricket_commentary_dataset/lf
cricket_commentary_dataset/nk
cricket_commentary_dataset/nl
cricket_commentary_dataset/oa
cricket_commentary_dataset/or
cricket_commentary_dataset/or
cricket_commentary_dataset/oz
```

Question 4: Find out the total free disk space on your system filesystems. The 'df -m' command will list all filesystem on your PC and the available column will mention the free space on each filesystem in megabyte. You can pipe this output to other commands and get the total free space, which will be the sum of the 'Available' column of all filesystems.

Correct solutions:

a. df -m | awk '{p+=$4}; END {print p}'
b. df -m | awk '{print $4}' | paste -s -d+ - | cut -d '+' -f 2- | bc
c. df -m | sed 's/ [ ]*/\t/g' | cut -f 4 | paste -s -d+ - | cut -d '+' -f 2- | bc
d. Any other command that gives the correct answer.

Question 5: How will you find all files larger than 1 mb present in the mid-semester-exam directory?

Correct solutions:

a. find . -size +1M
b. Any other command that gives the correct answer.

Question 6: The file 'alice.txt' is a text version of the book "Alice's Adventures in Wonderland". Find out the top 40 words based on their occurrence frequency used in the book. For simplicity assume words are delimited by spaces, which implies "Author:", "Gutenberg-tm," are valid words, but "which has" is not.

Correct solutions:

a. cat alice.txt | sed 's/ [ ]*/\n/g' | sort | uniq -c | sort -nr | head -n 40
b. cat alice.txt | sed -E 's/[ ]+/\n/g' | sort | uniq -c | sort -nr | head -n 40
c. Any other command that gives the correct answers.

Question 7:Network Time Protocol (NTP) is used to synchronize the clock of a local computer with a remote machine. If a system is connected to an unrestricted internet, usually all modern OS like Ubuntu will automatically update its date and time. However, if a local computer is behind a restricted internet access it is not possible for an OS to use NTP protocol to update date and time. Suppose your computer is behind such a restricted network, how will you update your system time automatically?

Hint: You can make a https request to google.com, the response received will contain a time stamp set by google servers. Assuming that google servers have up to date & time and the network latency is negligible, you can use this time stamp to update your system time. For this question, a sample response from google.com is available in google_response.txt file. You can extract the date and time values from this file. The expected answer should be in the format:
cat google_response.txt | multile_one_liner_commands
with the expected output: Tue Sep 10 16:27:18 GMT 2019.
Note that the output format of date and time is different from what is present in the text file and you have to get the output in the expected format.

Correct solutions:

a. cat google_response.txt | grep -E '^[[:space:]]*[dD]ate:' | sed 's/^[[:space:]]*[dD]ate:[[:space:]]*//' | head -1l | awk '{print $1, $3, $2, $5 ,"GMT", $4 }' | sed 's/,//'
b. Any other command that gives the correct answer.

Question 8: Suppose you are a system administrator of iitg.ac.in website. IITG website is hosted using Apache web server which maintains an access in a log file named apache.log. The director asked you to find out different statistics related to the website as described below:

a. How many unique visits did the website got during 7th december to 10th December 2010 (Both days are inclusive)? Multiple visits from the same IP will not be counted as unique visit.
b. FInd out the top 10 referrer webpages to the iitg website on 6th December?
c. What is the most popular web page accessed by the users.

Hint: The log description is as follows. One line contains a single request received by the server. Note that the data is not an actual data from iitg.ac.in but a real data from a website which also uses apache.

127.0.0.1 - - [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I ;Nav)"

The blue color string is the referrer webpage.
The black color string is IP address of the visitor.
The dark green color string is date and time.
The magenta color string is the web page accessed by the user.

Correct solutions:

a. cat apache.log | awk '{print $1}' | sort | uniq | wc -l
b. cat apache.log | grep -E '06/Dec|07/Dec|08/Dec|09/Dec|10/Dec' | awk -F\" '{print $4}' | sort | uniq -c | sort -nr | head -n 10
c. cat apache.log | awk '{print $7}' | sort | uniq -c | sort -nr | head -n 1
d. Any other command that gives the correct answer.