1.Create folder 'Test' in your home directory

**mkdir Test**

2.Create below files​

employeelist​ **- cat >employeelist**

skillset **- cat >skillset**

3. Create folder Test2 in home directory​

**mkdir Test2**

1. create symbolink and hard link for skillset file ​

**Symbolink: ln -s skillset s\_skillset**

**Hardlink: ln skillset h\_skillset**

1. delete skillset file and check if hardlink and symbolic link file exists​

**Upon deleting skillset file, symbolink file doesnt exist but hardlink file still exist.**

1. explain the difference b/n symbolic and hard links

**Hard link:** Hard link are associating two or more file names with the same node. you can create one or more hard links for a single file. Hard links cannot be created for directories and files on a different filesystem or partition.

**Symbolic link:** Its something like a shortcut in windows and it is an indirect pointer to a file or a directory on a different filesystem or partition.

4. Provide read and write permission to 'user2' group test if user2 group users can access the file

**chmod g+r skillset**

5. Revoke write permission from user2 group and test the access again

**chmod g-w skillset**

6. Grant write permission for others and test it

**chmod o+w skillset**

7. Revoke write permission for others and test again

**chmod o-w skillset**

8. Change the group of employeelist file to 'user3' test if user3 group have access

**chgrp user3 employeelist**

9. Change ownership of the file to user5 and test

**chown user5 skillset**

10. check how many employees are there  (hint: use wc command)

**wc employees -l**

11. Create file “File1”

**cat > File1**

12. Append 2 more lines to the same file

**cat >> File1**

13. Create File2 with few lines

**cat > File 2**

14. Display the contents of both File1 and File2

**cat File1**

**cat File2**

15. Concatenate both File1 & 2

**cat File1 File2**

16. Send the above output to File3

**cat File1 File2 >File3**

17. Read File1,File2,File3… File5 . Observe the output

**cat File1 File2 File3 File4 File5**

18. Redirect the errors of the above command to “errorlog”

**cat File1 File2 File3 File4 File5 2>errorlog**

19. Concatenate name File1 to 5 and store it in File 10.if any errors, log them in the same file

**cat File1 File2 File3 File4 File5 >File10 2>&1**

20.Copy File1 to File6.Add the errors to “errorlog”

**cp File1 File6 2>&errorlog**

21.Insert an heading at the beginning of the file

**Press O**

22.Remove the second line in the file and try undo command

**Place the cursor on second line.**

**Press “dd”**

**For undo press “u”**

23.Search for all occurences of bc

**:/bc**

24.Remove the word number

**Type :/number, then type dw on each word occurrence.**

25.Replace all occurences of char c in line1 only with C

**:l,s/c/C/g**

26. Replace all occurences of space with \t

**:1,s/ /\t/g**

Page no:**55**

27. The 10th to 25th lines in a file

**head -n +25 File1 | tail -15**

28. List only 10th line in a file

**head -n +10 File11 | tail -1**

26. List only the recently modified file in the current directory

**find . -type -ctime -1 -ls**

27. List only the smallest file in the current directory

**ls -lsr**

28. A data file is updated regularly, each day with 10 new records. New records are appended at the end. Write the command to print total number of records in the file.

**grep -c ".\*" File11**

29.Write the command to list only the records entered 3 days before.

**find . -iname "\*" -mtime +3 -exec cat {} \;**

30. Lines containing A

**grep "A" file**

31. lines containing The at the beginning

**grep ^The file**

32. lines ending with .

**grep "\.$" file**

33. lines with a,b or c as the second letter

**grep "^.a\|b\|c" File11**

34. lines which contain def , deef or deeeef

**grep "def\|deef\|deeeef" File**

35. lines not having numbers at the beginning

**grep -v [0-9] File**

36. empty lines

**grep -n "^$" File**

37. list only directories

**ls -d \*/**

38. Sort the file based on the filename descending order (last field)

**ls | sort -r**

39. Sort the file based only on the inode count

**ls -li | sort -n**

40. Sort the file based on the user name and group name in reverse order

**ls -lr | sort -k 3**

**ls -lr | sort -k 4**

41.Sort the file based on the file size and store the output in a file called sortedfile

**ls -laS >sortedfile**

**cat sortedfile**