1. ls
2. chmod
3. cat
4. du
5. ftp
6. ps
7. head
8. df
9. tail
10. who
11. tr
12. w
13. sort
14. kill
15. cut
16. fg
17. paste
18. arp
19. diff
20. nl
21. comm.
22. bg
23. grep
24. wc
25. uniq
26. rm
27. telnet
28. rmdir
29. egrep
30. chgrp
31. fgrep
32. chown

1) Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2) Write a shell script that deletes all lines containing a specified word in one or more files supplied as an argument.
3) Write a shell script that displays a list of files in the current directory to which the user has read, write and execute permissions.
4) Write a C program to list for every file in a directory, its inode number and file name.
5) Write an awk script to find the number of characters, words and lines in a file.
6) Write an awk script to count the number of lines in a file that do not contain vowels.
7) Write a shell script to find factorial of a given integer. And to list all the directory files in the given directory.
8) Write a C program that makes a copy of a file using standard I/O and system calls.
9) Write a shell script that receives a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
10) Write a C Program to implement cat command using system calls.
11) Write a C Program to implement mv command using system calls.
12) Write a C Program to implement ls command using system calls.
13) Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines in it is also reported.
14) Write a C program that illustrates communication between two unrelated processes using named pipe.
15) Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.
16) Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (21)) and displays them.
17) Write a C Program to create a zombie process.