Mini projects on Shell Scripting:

1) Book Rental Application

The ABC Book Company needs a way to determine the cost that a student has to pay for renting a Book.

The cost is dependent on the time of the Book is returned.

However, there are also special rates on Saturday and Sundays.

The Fee Structure is shown in the following list:

1) The cost is Rs 30 /- Per Day.
2) If the Book is returned after 9 PM, then student will be charged an extra day.
3) If the Book is rented on a Sunday, the student will get 50% off for as long as they keep the book.
4) If the Book is rented on a Saturday, the student will get 30% off as long as they keep the book.

We need to write a script to meet this requirement.

2) Book Management Application

Requirements:

1) Librarian can add new book information.
   It consists of book title, authors, price, ISBN, publication, edition and no. of copies.

2) Librarian can search a specific book by title or ISBN number.

3) Librarian can update book information.

4) Librarian can delete the specific book record.

(Simple CRUD operations for Book Management Application)

3) User Management Application

Requirements:

1) Add a new user -> Store the information like first name, last name, user id and password.
   (If user provides user id that already exists, then ask to provide different user id)

   (Check if password and confirm password is same or not)
   (Make sure that password is not visible on console)

2) Search a specific user by user id, if user is not found, give appropriate message.

3) User can change the password. Old password is required. Validate if old password is correct or not. Also validate password and confirm password.

4) Delete user.

5) Display all user information.

++

1) Write a shell script which receives two filenames as arguments. It should check whether the two file’s content are same or not. If they are same then second file should be deleted.
   (Hint: use cmp command to compare files)

2) While executing a shell script either LOGNAME or UID is supplied at the command prompt. Write a shell script to find out at how many terminals has this user logged in.

3) Write a shell script, which gets executed the moment user logs in. It should display the message. “Good morning”/”Good Afternoon”/”Good Evening”/”Good Night” depending upon the time at which user logs in.

4) Write a menu driven program which has following options:

   1. Contents of /etc/passwd
   2. List of users who have currently logged in
   3. Present working directory
   4. Exit

Make use of case statement.

5) Write a script to calculate overtime pay of 10 employees. Overtime is paid at the rate of Rs. 1200/- per hour for every hour worked above 40 hours. Assume that employee don’t work for fractional part of an hour.

6) Write a shell script which reports names and sizes of all files in directory (directory would be supplied as an argument to shell script) whose size is exceeding 1000 bytes. The filenames should be printed in descending order of their sizes. The total number of such files should also be reported.

7) Write a shell script which displays a list of all files in the current directory to which you have read, write and execute permissions.

Construct pipeline to carry out the following tasks:

8) List of all files beginning with the character “p” on the screen and also store them in file called file1.
9) Output of who command should be sorted and displayed on screen along with total number of users. The same output except the number of users should also be sorted in file called users.
10) Merge the contents of the files a.txt, b.txt and c.txt, sort them and display the sorted output on the screen page by page.
11) Display the list of last 20 files present in current directory. Also store this list in file called file2.

Perform the following operations:

1) Create 5 empty files file1, file2, file3, file4, file5
2) Create a file called data.txt and store your name, age, address, email, gender and contact in it.
3) Display the contents of file data.txt on the screen.
4) Make copy of data.txt into another file data_backup.txt
5) Create a temp.txt file and write any 2 sentences in it.
6) Combine the content of data.txt and temp.txt into another file tempdata.txt.
7) Delete the file data.txt

8) Create one link called tempfilelink.txt for the file temp.txt
9) Change the permission for the file data_backup.txt to 666
10) Rename the file data_backup.txt to data.txt
11) Create a directory called mydir in current directory
12) Move the files data.txt and tempdata.txt in mydir
13) Create another directory newdir under mydir
14) Copy the contents of mydir to newdir
15) Delete the mydir and newdir in one shot.

++

1) Create a file data.txt under /opt.

   a. Alex and Peter able to read and execute
   b. Harry able to read and execute
   c. Remaining users able to read (For eg. Tom…)
   d. Bob able to read and execute

2) Create a file notes.txt and give full permission via octal way.

3) Create a file test with rw-rwx---

4) Create a file myfile with r-- r--r—via octal way

5) Create a file myfile under /home and create hard link.

6) Create a shortcut of myfile under /var

7) Delete parent file i.e. myfile and check orphan files.

8) Create a new user Sachin and try to switch user su – Sachin

9) Get a total count of files/symlinks/directories from OS.