1. Write a shell script which will generate the O/P as follows

\*

\*\*

\*\*\*

\*\*\*\*

Ans:

i=1

while [ $i -le 4 ]

do

  j=1

  while [ $j -le $i ]

  do

    echo -n "\*"

    j=$((j + 1))

  done

  echo

  i=$((i + 1))

done

1. Accept the first name, middle name, and last name of a person in variables fname, mname and lname respectively. Greet the person (take his full name) using appropriate message.

read -p "Enter your first name: " name

read -p "Enter your middle name: " middle

read -p "Enter your last name: " surname

echo "Hello, $name $middle $surname! Welcome!"

1. Display the name of files in the current directory along with the names of files with maximum & minimum size. The file size is considered in bytes.

echo "Files in the directory:"

ls -p | grep -v /

max\_file=$(ls -S | head -n 1)

min\_file=$(ls -Sr | head -n 1)

echo "maximum size: $max\_file"

echo "minimum size: $min\_file"

1. Write a script which when executed checks out whether it is a working day or not?

(Note: Working day Mon-Fri)

day=$(date +%u)

if [ $day -ge 1 ] && [ $day -le 5 ]; then

  echo "It is a working day."

else

  echo "holiday"

1. Write a script that accepts a member into HP health club, if the weight of the person is withing the range of 30-250 Kgs.

read -p "Enter your weight in kg: " weight

if [ $weight -ge 30 ] && [ $weight -le 250 ]; then

  echo "welcome to HP Health Club."

else

  echo " you are fat“

1. Write a shell script that greets the user with an appropriate message depending on the system time.

hour=$(date +%H)

minute=$(date +%M)

second=$(date +%S)

current\_time="$hour:$minute:$second"

echo "Welcome, the current time is $current\_time."

1. A data file file has some student records including rollno, names and subject marks. The fields are separated by a “:”. Write a shell script that accepts roll number from the user, searches it in the file and if the roll number is present - allows the user to modify name and marks in 3 subjects.   
   If the roll number is not present, display a message “Roll No Not Found”. Allow the user to modify one record at a time.

read -p "Enter the roll number to search: " rollno

record=$(grep "^$rollno:" file)

if [ -z "$record" ]; then

  echo "Roll No Not Found"

else

  echo "Current record: $record"

  read -p "Enter  name: " name

  read -p "marks for Subject 1: " sub1

  read -p "marks for Subject 2: " sub2

  read -p "marks for Subject 3: " sub3

1. Modify program 7 to accept the RollNo from the command line.

rollno=$1

if [ -z "$rollno" ]; then

  echo " roll number."

  exit 1

fi

if [ ! -f "file" ]; then

  echo "File not found!"

  exit 1

fi

record=$(grep "^$rollno:" file)

if [ -z "$record" ]; then

  echo "Roll No Not Found"

else

  echo "Current record: $record"

  read -p "Enter new name: " name

  read -p "Enter marks for Subject 1: " sub1

  read -p "Enter marks for Subject 2: " sub2

  read -p "Enter marks for Subject 3: " sub3

1. Modify the program 7 to accept the RollNo and display the record and ask for delete confirmation. Once confirmed delete the record and update the data file.

rollno=$1

if [ -z "$rollno" ]; then

  echo " roll number."

  exit 1

fi

record=$(grep "^$rollno:" file)

if [ -z "$record" ]; then

  echo "Roll No Not Found"

else

  echo "Current record: $record"

1. Write a script that takes a command line argument and reports on its file type (regular file, directory file, etc.). For more than one argument generate error message.

if [ $# -ne 1 ]; then

  echo "Error: Please provide exactly one argument."

  exit 1

fi

file=$1

if [ -e "$file" ]; then

  if [ -f "$file" ]; then

    echo "$file is a regular file."

  elif [ -d "$file" ]; then

    echo "$file is a directory."

  elif [ -l "$file" ]; then

    echo "$file is a symbolic link."

  else

    echo "$file is some other type of file."

  fi

else

  echo "$file does not exist."

fi

1. Add some student records in the “student” file manually. The fields to be considered are “RollNo”, “Name”, “Marks\_Hindi”, “Marks\_Maths”, “Marks\_Physics”.  
    Write a script which does the following
   1. If the roll number already exists, then store the record and the following message   
      “roll number exists” in a log file “log1”.
   2. If the marks in the subjects is not in the range of 1 – 99 then store such a record followed by a message “marks out of range” in “log1”
   3. If the data is valid, the calculate total, percentage, grade and display on the terminal

read -p "Enter Roll Number: " rollno

read -p "Enter Name: " name

read -p "Enter Marks in Hindi: " hindi

read -p "Enter Marks in Maths: " maths

read -p "Enter Marks in Physics: " physics

if grep -q "^$rollno:" student; then

  echo "Roll No $rollno: $name, Hindi: $hindi, Maths: $maths, Physics: $physics" >> log1

  echo "Roll number exists. The record has been logged in log1."

else

  echo "$rollno:$name:$hindi:$maths:$physics" >> student

if [ "$hindi" -lt 1 ] || [ "$hindi" -gt 99 ] || [ "$maths" -lt 1 ] || [ "$maths" -gt 99 ] || [ "$physics" -lt 1 ] || [ "$physics" -gt 99 ]; then

  echo "Roll No $rollno: $name, Hindi: $hindi, Maths: $maths, Physics: $physics - marks out of range" >> log1

  echo "Marks out of range. The record has been logged in log1."

else

  total=$((hindi + maths + physics))

  percentage=$((total / 3))

  echo "Total Marks: $total"

  echo "Percentage: $percentage%"