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

\*

\*\*

\*\*\*

\*\*\*\*

for row in {1..4}

do

for column in $(seq 1 $row)

do

echo -n "\*"

done

echo ""

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: " fname

read -p "Enter your middle name: " mname

read -p "Enter your last name: " lname

echo "Hello, $fname $mname $lname! 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 current directory:"

for file in \*

do

if [ -f "$file" ]; then

echo "$file"

fi

done

# Find the file with the maximum size

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

# Find the file with the minimum size

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

echo ""

echo "File with the maximum size: $max\_size\_file"

echo "File with the minimum size: $min\_size\_file"

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

(Note: Working day Mon-Fri)

current\_day=$(date +%u)

if [ $current\_day -ge 1 ] && [ $current\_day -le 5 ]; then

echo "It's a working day."

else

echo "It's not a working day."

fi

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 "Sorry, your weight does not meet the requirements for membership."

fi

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

current\_hour=$(date +%H)

if [ $current\_hour -ge 5 ] && [ $current\_hour -lt 12 ]; then

greeting="Good Morning!"

elif [ $current\_hour -ge 12 ] && [ $current\_hour -lt 17 ]; then

greeting="Good Afternoon!"

elif [ $current\_hour -ge 17 ] && [ $current\_hour -lt 21 ]; then

greeting="Good Evening!"

else

greeting="Good Night!"

fi

echo "$greeting"

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.

data\_file="student\_records.txt"

read -p "Enter Roll Number: " roll\_number

record=$(grep "^$roll\_number:" $data\_file)

if [ -n "$record" ]; then

echo "Record Found: $record"

read -p "Enter New Name: " new\_name

read -p "Enter New Marks for Physics: " new\_marks1

read -p "Enter New Marks for Chemistry: " new\_marks2

read -p "Enter New Marks for Maths: " new\_marks3

new\_record="$roll\_number:$new\_name:$new\_marks1:$new\_marks2:$new\_marks3"

sed -i "s/^$roll\_number:.\*/$new\_record/" $data\_file

echo "Record Updated Successfully."

else

echo "Roll No Not Found"

fi

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

data\_file="student\_records.txt"

if [ -z "$1" ]; then

echo "Usage: $0 <RollNumber>"

exit 1

fi

roll\_number=$1

record=$(grep "^$roll\_number:" $data\_file)

if [ -n "$record" ]; then

echo "Record Found: $record"

read -p "Enter New Name: " new\_name

read -p "Enter New Marks for Physics: " new\_marks1

read -p "Enter New Marks for Chemistry: " new\_marks2

read -p "Enter New Marks for Maths: " new\_marks3

new\_record="$roll\_number:$new\_name:$new\_marks1:$new\_marks2:$new\_marks3"

sed -i "s/^$roll\_number:.\*/$new\_record/" $data\_file

echo "Record Updated Successfully."

else

echo "Roll No Not Found"

fi

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.

data\_file="student\_records.txt"

if [ -z "$1" ]; then

echo "Usage: $0 <RollNumber>"

exit 1

fi

roll\_number=$1

record=$(grep "^$roll\_number:" $data\_file)

if [ -n "$record" ]; then

echo "Record Found: $record"

read -p "Do you really want to delete this record? (y/n): " confirmation

if [ "$confirmation" = "y" ]; then

sed -i "/^$roll\_number:/d" $data\_file

echo "Record Deleted Successfully."

else

echo "Record Deletion Cancelled."

fi

else

echo "Roll No Not Found"

fi

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 [ -f "$file" ]; then

echo "$file is a regular file."

elif [ -d "$file" ]; then

echo "$file is a directory."

else

echo "$file is of an unknown type."

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

data\_file="student"

log\_file="log1"

read -p "Enter Roll Number: " roll\_no

read -p "Enter Name: " name

read -p "Enter Marks in Hindi: " marks\_hindi

read -p "Enter Marks in Maths: " marks\_maths

read -p "Enter Marks in Physics: " marks\_physics

# Check if roll number already exists

if grep -q "^$roll\_no:" $data\_file; then

echo "$roll\_no:$name:$marks\_hindi:$marks\_maths:$marks\_physics - roll number exists" >> $log\_file

echo "roll number exists"

exit 1

fi

# Check if marks are within range

if [ $marks\_hindi -lt 1 ] || [ $marks\_hindi -gt 99 ] || [ $marks\_maths -lt 1 ] || [ $marks\_maths -gt 99 ] || [ $marks\_physics -lt 1 ] || [ $marks\_physics -gt 99 ]; then

echo "$roll\_no:$name:$marks\_hindi:$marks\_maths:$marks\_physics - marks out of range" >> $log\_file

echo "marks out of range"

exit 1

fi

# Calculate total, percentage, and grade

total=$(($marks\_hindi + $marks\_maths + $marks\_physics))

percentage=$(($total / 3))

if [ $percentage -ge 90 ]; then

grade="A"

elif [ $percentage -ge 75 ]; then

grade="B"

elif [ $percentage -ge 60 ]; then

grade="C"

elif [ $percentage -ge 50 ]; then

grade="D"

else

grade="F"

fi

# Display the results

echo "Total Marks: $total"

echo "Percentage: $percentage%"

echo "Grade: $grade"

echo "$roll\_no:$name:$marks\_hindi:$marks\_maths:$marks\_physics" >> $data\_file