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

\*

\*\*

\*\*\*

\*\*\*\*

**# vim pattern.sh**

**#!/bin/bash**

**for ((i=1;i<=4;i++)); do**

**for ((j=1;j<=i;j++); do**

**echo -n “\*”**

**done**

**echo “”**

**done**

**# chmod +x pattern.sh**

**# ./pattern.sh**

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.

**# vim message.sh**

**#!/bin/bash**

**read -p “Enter First Name: “ fname**

**read -p “Enter Middle Name: “ mname**

**read -p “Enter Last Name: “ lname**

**echo “Hello, $fname $mname $lname! Welcome!.”**

**chmod +x message.sh**

**./message.sh**

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.

**#vim file\_sizes.sh**

**#!/bin/bash**

**echo “Files in the current directory:”**

**ls -lS -block-size=1 | awk ‘{print $5, $9}’ | tail -n +2**

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

**max\_size=$(stat -c%s “$max\_file”)**

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

**max\_size=$(stat -c%s “$min\_file”)**

**echo “File with the maximum size: $max\_file ($max\_size bytes)”**

**echo “File with the maximum size: $min\_file ($min\_size bytes)”**

**# chmod +x file\_sizes.sh**

**# ./file\_sizes.sh**

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

(Note: Working day Mon-Fri)

**# vim working.sh**

**#!/bin/bash**

**day=$(date +%a)**

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

**echo “It’s working Day”**

**else**

**echo “It’s weekend”**

**fi**

**# chmod +x working.ssh**

**# ./working.ssh**

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.

**# vim health\_club.sh**

**#!bin/bash**

**read -p “Enter Weight:” weight**

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

**echo “Welcome to club”**

**else**

**echo “Weight not Acceptable”**

**fi**

**# chmod +x health\_club**

**# ./health\_club**

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

**#vim greet.sh**

**#!bin/bash**

**hour=$(date +%H)**

**if [“$hour” -ge 5]&& [“$hour” -lt 12]; then**

**echo “Good Morning”**

**elif [“$hour” -ge 12]&& [“$hour” -lt 18]; then**

**echo “Good Afternoon”**

**elif [“$hour” -ge 18]&& [“$hour” -lt 22]; then**

**echo “Good Evening”**

**else**

**echo “Good Night”**

**fi**

**# chmod +x greet.sh**

**# ./greet.sh**

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.

**#vim update.sh**

**#!bin/bash**

**file=”students.txt”**

**read -p “Enter roll no:” rollno**

**record=$grep “^$rollno:” “$file”)**

**if [-n “$record” ]; then**

**echo “Roll No not found”**

**else**

**echo “Record found: $record”**

**read -p “Enter new name (current: $current\_name): “new\_name**

**read -p “Enter new marks for Subject1 (current: $current\_marks1): “new\_marks1**

**read -p “Enter new marks for Subject2 (current: $current\_marks2): “new\_marks2**

**read -p “Enter new marks for Subject3 (current: $current\_marks3): “new\_marks3**

**new\_record=”$roll\_no:$new\_name:$new\_marks1:$new\_marks2:$new\_marks3”**

**sed -i ”s|^$record|$new\_record| “$file”**

**echo “Record updated successfully”**

**fi**

**# chmod +x update.sh**

**# ./update.sh**

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

**#vim update.sh**

**#!bin/bash**

**file=”students.txt”**

**if [-z “$1” ]; then**

**echo “Usage: $0 <Roll Number”**

**exit 1**

**fi**

**roll\_no=”$1”**

**record=$(grep “^$roll\_no:” “$file”)**

**if [-z “$record” ]; then**

**echo “Roll No not found”**

**else**

**echo “Record found: $record”**

**read -p “Enter new name (current: $current\_name): “new\_name**

**read -p “Enter new marks for Subject1 (current: $current\_marks1): “new\_marks1**

**read -p “Enter new marks for Subject2 (current: $current\_marks2): “new\_marks2**

**read -p “Enter new marks for Subject3 (current: $current\_marks3): “new\_marks3**

**new\_record=”$roll\_no:$new\_name:$new\_marks1:$new\_marks2:$new\_marks3”**

**sed -i ”s|^$record|$new\_record| “$file”**

**echo “Record updated successfully”**

**fi**

**# chmod +x update.sh**

**# ./update.sh**

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.

**#vim update.sh**

**#!bin/bash**

**file=”students.txt”**

**if [-z “$1” ]; then**

**echo “Usage: $0 <Roll Number”**

**exit 1**

**fi**

**roll\_no=”$1”**

**record=$(grep “^$roll\_no:” “$file”)**

**if [-z “$record” ]; then**

**echo “Roll No not found”**

**else**

**echo “Record found: $record”**

**read -p “Do you want to delete this record?(y/n):” confirm**

**if [[“$confirm” == ”y” || “$confirm" == “Y” ]]; then**

**sed -i “/^$roll\_no:/d” “$file”**

**echo “Record Deleted”**

**else**

**echo “Deletion Cancelled”**

**fi**

**fi**

**# chmod +x update.sh**

**# ./update.sh**

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.

**#!bin/bash**

**if [ “$#” -ne 1 ]; then**

**echo “Usage: $0 <filename>”**

**exit 1**

**fi**

**file=”$1”**

**if [ -e “$file” ]; then**

**if [ -f “$file” ]; then**

**echo “$file is regular file”**

**elif [ -d “$file” ]; then**

**echo "$file is directory”**

**else**

**echo “$file is unknown type.”**

**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

**# vim student.sh**

**#!bin/bash**

**file=”students.txt”**

**log=”log1”**

**read -p “Enter Roll no: “rollno**

**read -p “Enter name: “name**

**read -p “Enter marks for Subject1: “marks1**

**read -p “Enter marks for Subject2: “marks2**

**read -p “Enter marks for Subject3: “marks3**

**record=”$rollno:$name:$marks1:$marks2:$marks3”**

**if grep -q “^$rollno:” “$file”; then**

**echo “$record roll number exists” >> “$log”**

**echo “Roll Number exists.Logged to $log.”**

**elif [[ $marks1 -lt 1 || $marks1 -gt 99 || $marks2 -lt 1 || $marks2 -gt 99 || $marks3 -lt 1 || $marks3 -gt 99 ]]; then**

**echo “$rollno:$name:$marks1:$marks2:$marks3” >> “log”**

**echo “$record marks out of range” >> “$log”**

**echo “Marks out of Range. Logged to $log.”**

**else**

**total=$((marks1 + marks2 + marks3))**

**percentage=$((total \*100 / 3))**

**if [ “$percentage” -ge 90 ]; then**

**grade=”A”**

**elif [ “$percentage” -ge 75 ]; then**

**grade=”B”**

**elif [ “$percentage” -ge 50 ]; then**

**grade=”C”**

**else**

**grade=”D”**

**fi**

**echo “$record:$total:$percentage:$grade” >>”$file”**

**echo “Total=$total, Percentage=$percentage%, Grade=$grade”**

**fi**

**# chmod +x sudent.sh**

**# ./student.sh**