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

\*

\*\*

\*\*\*

**Ans:** vim stars.sh

#!/bin/bash

for i in {1..4}

do

for j in $(seq 1 $i)

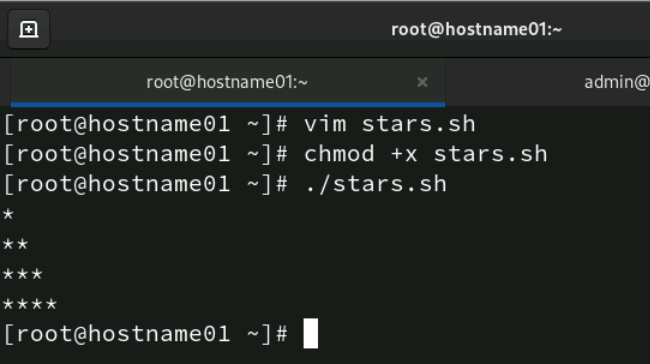
do

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.

Ans:

vim greet.sh

#!/bin/bash

# Accept the first name, middle name, and last name

echo "Enter your first name:"

read fname

echo "Enter your middle name:"

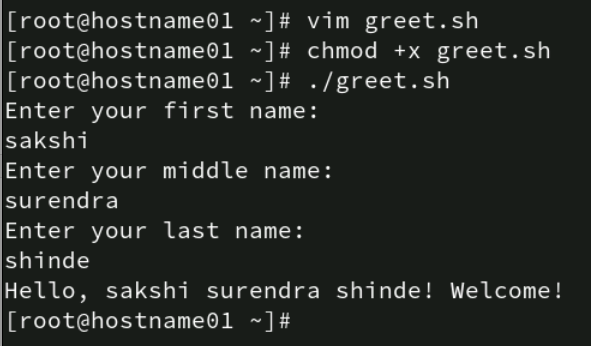
read mname

echo "Enter your last name:"

read lname

# Greet the person

echo "Hello, $fname $mname $lname! Welcome!"



3.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.

Ans: vim file\_sizes.sh

#!/bin/bash

# List all files in the current directory

echo "Files in the current directory:"

ls -1

# Find the file with the maximum size

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

max\_size=$(stat -c%s "$max\_file")

# Find the file with the minimum size

min\_file=$(ls -S | tail -n 1)

min\_size=$(stat -c%s "$min\_file")

# Display the file with the maximum size

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

# Display the file with the minimum size

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

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

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

(Note: Working day Mon-Fri)

**Ans:vim check\_workingdays.sh**

**#!/bin/bash**

**# Get the current day of the week (1-7, where 1 is Monday and 7 is Sunday)**

**day\_of\_week=$(date +%u)**

**# Check if the day is a working day (Monday to Friday)**

**if [ "$day\_of\_week" -ge 1 ] && [ "$day\_of\_week" -le 5 ]; then**

**echo "Today is a working day."**

**else**

**echo "Today is not a working day."**

**fi**

A screenshot of a computer

Description automatically generated

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.

**Ans: vim health\_club**

**#!/bin/bash**

**# Prompt the user to enter their weight**

**echo "Enter your weight in kg:"**

**read weight**

**# Check if the weight is within the acceptable range**

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

**echo "Welcome to the HP health club!"**

**else**

**echo "Sorry, your weight is not within the acceptable range for membership."**

**fi**

A computer screen shot of white text

Description automatically generated

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

**Ans: vim greet\_user.sh**

**#!/bin/bash**

**# Get the current hour (0-23)**

**current\_hour=$(date +%H)**

**# Determine the appropriate greeting based on the current hour**

**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"**

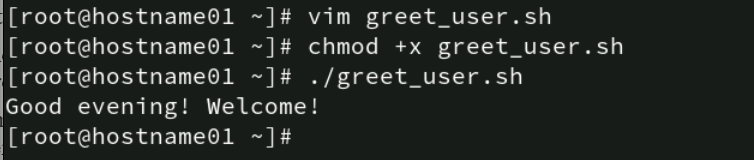
**elif [ "$current\_hour" -ge 17 ] && [ "$current\_hour" -lt 21 ]; then**

**greeting="Good evening"**

**else**

**greeting="Good night"**

**# Greet the user**

**echo "$greeting! Welcome!”**

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.

**Ans: vim update\_student.sh**

**#!/bin/bash**

**# File containing student records**

**data\_file="students.txt"**

**# Prompt the user to enter the roll number**

**echo "Enter the roll number:"**

**read rollno**

**# Search for the roll number in the file**

**record=$(grep "^$rollno:" "$data\_file")**

**if [ -z "$record" ]; then**

**echo "Roll No Not Found"**

**else**

**# Display the current record**

**echo "Current record: $record"**

**# Prompt the user to enter the new name and marks**

**echo "Enter the new name:"**

**read new\_name**

**echo "Enter the new marks for subject 1:"**

**read new\_marks1**

**echo "Enter the new marks for subject 2:"**

**read new\_marks2**

**echo "Enter the new marks for subject 3:"**

**read new\_marks3**

**# Create the new record**

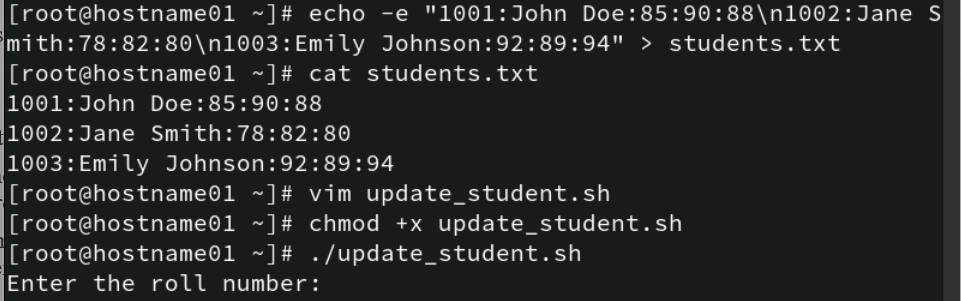
**new\_record="$rollno:$new\_name:$new\_marks1:$new\_marks2:$new\_marks3"**

**# Replace the old record with the new record in the file**

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

**echo "Record updated successfully."**

**fi**



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

**Ans: vim update\_student.sh**

**#!/bin/bash**

**# File containing student records**

**data\_file="students.txt"**

**# Check if the roll number is provided as a command-line argument**

**if [ -z "$1" ]; then**

**echo "Usage: $0 <rollno>"**

**exit 1**

**fi**

**# Get the roll number from the command-line argument**

**rollno=$1**

**# Search for the roll number in the file using grep**

**record=$(grep "^$rollno:" "$data\_file")**

**if [ -z "$record" ]; then**

**echo "Roll No Not Found"**

**else**

**# Display the current record**

**echo "Current record: $record"**

**# Prompt the user to enter the new name and marks**

**echo "Enter the new name:"**

**read new\_name**

**echo "Enter the new marks for subject 1:"**

**read new\_marks1**

**echo "Enter the new marks for subject 2:"**

**read new\_marks2**

**echo "Enter the new marks for subject 3:"**

**read new\_marks3**

**# Create the new record**

**new\_record="$rollno:$new\_name:$new\_marks1:$new\_marks2:$new\_marks3"**

**# Replace the old record with the new record in the file using sed**

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

**echo "Record updated successfully."**

**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.

**Ans: 1.vim delete\_student.sh**

**#!/bin/bash**

**# File containing student records**

**data\_file="students.txt"**

**# Check if the roll number is provided as a command-line argument**

**if [ -z "$1" ]; then**

**echo "Usage: $0 <rollno>"**

**exit 1**

**fi**

**# Get the roll number from the command-line argument**

**rollno=$1**

**# Search for the roll number in the file using grep**

**record=$(grep "^$rollno:" "$data\_file")**

**if [ -z "$record" ]; then**

**echo "Roll No Not Found"**

**else**

**# Display the current record**

**echo "Current record: $record"**

**# Ask for delete confirmation**

**echo "Do you want to delete this record? (yes/no)"**

**read confirmation**

**if [ "$confirmation" = "yes" ]; then**

**# Delete the record from the file using sed**

**sed -i "/^$rollno:/d" "$data\_file"**

**echo "Record deleted successfully."**

**else**

**echo "Deletion cancelled."**

**fi**

**fi**

**2.chmod +x delete\_student.sh**

**3. ./delete\_student.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.

Ans:

**1] vim file\_type.sh**

**#!/bin/bash**

**# Check if more than one argument is provided**

**if [ "$#" -ne 1 ]; then**

**echo "Usage: $0 <filename>"**

**exit 1**

**fi**

**# Get the filename from the command-line argument**

**filename=$1**

**# Check if the file exists**

**if [ ! -e "$filename" ]; then**

**echo "File does not exist."**

**exit 1**

**fi**

**# Determine the file type**

**if [ -f "$filename" ]; then**

**echo "$filename is a regular file."**

**elif [ -d "$filename" ]; then**

**echo "$filename is a directory."**

**elif [ -L "$filename" ]; then**

**echo "$filename is a symbolic link."**

**else**

**echo "$filename is of another file type."**

**fi**

**2] chmod +x file\_type.sh**

**3] ./file\_type.sh <filename>**

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

Ans:

**vim student\_records.sh**

**#!/bin/bash**

**# File containing student records**

**data\_file="student"**

**log\_file="log1"**

**# Function to calculate grade based on percentage**

**calculate\_grade() {**

**local percentage=$1**

**if (( $(echo "$percentage >= 90" | bc -l) )); then**

**echo "A"**

**elif (( $(echo "$percentage >= 80" | bc -l) )); then**

**echo "B"**

**elif (( $(echo "$percentage >= 70" | bc -l) )); then**

**echo "C"**

**elif (( $(echo "$percentage >= 60" | bc -l) )); then**

**echo "D"**

**else**

**echo "F"**

**fi**

**}**

**# Prompt the user to enter student details**

**echo "Enter Roll Number:"**

**read rollno**

**echo "Enter Name:"**

**read name**

**echo "Enter Marks in Hindi:"**

**read marks\_hindi**

**echo "Enter Marks in Maths:"**

**read marks\_maths**

**echo "Enter Marks in Physics:"**

**read marks\_physics**

**# Check if the roll number already exists**

**if grep -q "^$rollno:" "$data\_file"; then**

**echo "$rollno:$name:$marks\_hindi:$marks\_maths:$marks\_physics" >> "$log\_file"**

**echo "roll number exists" >> "$log\_file"**

**echo "Roll number exists. Logged in $log\_file."**

**exit 1**

**fi**

**# Check if the marks are in the valid 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 "$rollno:$name:$marks\_hindi:$marks\_maths:$marks\_physics" >> "$log\_file"**

**echo "marks out of range" >> "$log\_file"**

**echo "Marks out of range. Logged in $log\_file."**

**exit 1**

**fi**

**# Calculate total, percentage, and grade**

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

**percentage=$(echo "scale=2; $total / 3" | bc)**

**grade=$(calculate\_grade "$percentage")**

**# Display the results**

**echo "Total Marks: $total"**

**echo "Percentage: $percentage%"**

**echo "Grade: $grade"**

**# Append the valid record to the data file**

**echo "$rollno:$name:$marks\_hindi:$marks\_maths:$marks\_physics:$total:$percentage:$grade" >> "$data\_file"**

**2] chmod +x student\_records.sh**

**3] ./student\_records.sh**