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

\*

\*\*

\*\*\*

\*\*\*\*

Code:

=>vi triangle.sh

#!/bin/bash

# Number of rows

rows=4

# Loop to print the pattern

for ((i=1; i<=rows; i++)); do

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

echo -n "\*"

done

echo "" # Move to the next line

done

chmod +x triangle.sh

./triangle.sh

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.

Code:

=>vi greet.sh

#!/bin/bash

# Accept first name, middle name, and last name

read -p "Enter First Name: " fname

read -p "Enter Middle Name: " mname

read -p "Enter Last Name: " lname

# Display Greeting Message

echo "Hello, $fname $mname $lname! Welcome to Shell Scripting."

chmod +x greet.sh

./greet.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.

Code:

=>vi file\_size.sh

#!/bin/bash

# Display all files with their sizes

echo "Files in the Current Directory:"

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

# Get the smallest and largest files

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

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

# Get their sizes in bytes

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

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

# Display Results

echo -e "\n📂 Largest File: $max\_file ($max\_size bytes)"

echo "📂 Smallest File: $min\_file ($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)

Code:

=>vi working\_day.sh

#!/bin/bash

# Get the current day (e.g., Mon, Tue, Wed, etc.)

day=$(date +%a)

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

if [[ "$day" == "Mon" || "$day" == "Tue" || "$day" == "Wed" || "$day" == "Thu" || "$day" == "Fri" ]]; then

echo "✅ It's a working day! Get to work."

else

echo "🎉 It's the weekend! Enjoy your break."

fi

chmod +x working\_day.sh

./working\_day.sh

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.

=>health\_club.sh

#!/bin/bash

# Accept the weight from the user

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

# Check if the weight is within the valid range

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

echo "✅ Welcome to HP Health Club! Your membership is accepted."

else

echo "❌ Sorry, your weight is outside the acceptable range (30-250 kg)."

fi

chmod +x health\_club.sh

./health\_club.sh

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

time.

Code:

=>health\_club.sh

#!/bin/bash

# Accept the weight from the user

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

# Check if the weight is within the valid range

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

echo "✅ Welcome to HP Health Club! Your membership is accepted."

else

echo "❌ Sorry, your weight is outside the acceptable range (30-250 kg)."

fi

chmod +x health\_club.sh

./health\_club.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.

Code:

greet\_time.sh

#!/bin/bash

# Get the current hour (24-hour format)

hour=$(date +%H)

# Greet based on the time of day

if [ "$hour" -ge 5 ] && [ "$hour" -lt 12 ]; then

echo "Good Morning! Have a great day ahead."

elif [ "$hour" -ge 12 ] && [ "$hour" -lt 18 ]; then

echo "Good Afternoon! Keep up the good work."

elif [ "$hour" -ge 18 ] && [ "$hour" -lt 22 ]; then

echo "Good Evening! Relax and unwind."

else

echo "Good Night! Sleep well and recharge."

fi

chmod +x greet\_time.sh

./greet\_time.sh

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

Code:

=>vi modify\_record.sh

#!/bin/bash

# File containing student records

file="file" # Replace with the actual filename if necessary

# Accept the roll number from the user

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

# Search for the roll number in the file

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

# If roll number exists in the file

if [ -n "$record" ]; then

echo "Record found: $record"

# Split the existing record into its components

IFS=':' read -r current\_rollno current\_name current\_marks1 current\_marks2 current\_marks3 <<< "$record"

# Accept the new name and marks for the 3 subjects

read -p "Enter new name (Current: $current\_name): " new\_name

read -p "Enter new marks for Subject 1 (Current: $current\_marks1): " new\_marks1

read -p "Enter new marks for Subject 2 (Current: $current\_marks2): " new\_marks2

read -p "Enter new marks for Subject 3 (Current: $current\_marks3): " new\_marks3

# Update the record in the file

sed -i "s/^$rollno:$current\_name:$current\_marks1:$current\_marks2:$current\_marks3$/$rollno:$new\_name:$new\_marks1:$new\_marks2:$new\_marks3/" "$file"

echo "Record updated successfully."

else

echo "Roll No Not Found"

fi

chmod +x modify\_record.sh

./modify\_record.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.

vi modify\_record\_cli.sh

#!/bin/bash

# File containing student records

file="file" # Replace with the actual filename if necessary

# Accept the roll number from the command line argument

if [ -z "$1" ]; then

echo "Error: Please provide a roll number as a command-line argument."

exit 1

fi

rollno=$1

# Search for the roll number in the file

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

# If roll number exists in the file

if [ -n "$record" ]; then

echo "Record found: $record"

# Split the existing record into its components

IFS=':' read -r current\_rollno current\_name current\_marks1 current\_marks2 current\_marks3 <<< "$record"

# Accept the new name and marks for the 3 subjects

read -p "Enter new name (Current: $current\_name): " new\_name

read -p "Enter new marks for Subject 1 (Current: $current\_marks1): " new\_marks1

read -p "Enter new marks for Subject 2 (Current: $current\_marks2): " new\_marks2

read -p "Enter new marks for Subject 3 (Current: $current\_marks3): " new\_marks3

# Update the record in the file

sed -i "s/^$rollno:$current\_name:$current\_marks1:$current\_marks2:$current\_marks3$/$rollno:$new\_name:$new\_marks1:$new\_marks2:$new\_marks3/" "$file"

echo "Record updated successfully."

else

echo "Roll No Not Found"

fi

chmod +x modify\_record\_cli.sh

./modify\_record\_cli.sh 105

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.

Code:

#!/bin/bash

# File containing student records

file="file" # Replace with the actual filename if necessary

# Accept the roll number from the command line argument

if [ -z "$1" ]; then

echo "Error: Please provide a roll number as a command-line argument."

exit 1

fi

rollno=$1

# Search for the roll number in the file

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

# If roll number exists in the file

if [ -n "$record" ]; then

echo "Record found: $record"

# Ask for confirmation to delete the record

read -p "Are you sure you want to delete this record? (y/n): " confirm

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

# Delete the record from the file

sed -i "/^$rollno:/d" "$file"

echo "Record deleted successfully."

else

echo "Deletion cancelled."

fi

else

echo "Roll No Not Found"

fi

./modify\_record\_cli.sh 105

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

Code:

vi student\_record.sh

#!/bin/bash

# File to store student records

file="student" # Replace with your file if different

logfile="log1" # Log file to store errors

# Accept student information

read -p "Enter Roll Number: " rollno

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 the roll number already exists in the file

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

echo "Roll number $rollno exists." >> "$logfile"

echo "Roll number exists. Record logged in $logfile."

exit 1

fi

# Validate marks to ensure they are within the range of 1-99

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:$marks\_physics - Marks out of range." >> "$logfile"

echo "Marks out of range. Record logged in $logfile."

exit 1

fi

# Calculate total, percentage, and grade

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

percentage=$((total \* 100 / 300))

# Determine the grade based on percentage

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

# Display the result on terminal

echo "Record for Roll No: $rollno"

echo "Name: $name"

echo "Marks in Hindi: $marks\_hindi"

echo "Marks in Maths: $marks\_maths"

echo "Marks in Physics: $marks\_physics"

echo "Total Marks: $total"

echo "Percentage: $percentage%"

echo "Grade: $grade"

# Save the valid record into the file

echo "$rollno:$name:$marks\_hindi:$marks\_maths:$marks\_physics" >> "$file"

chmod +x student\_record.sh

./student\_record.sh