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

#!/bin/bash

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

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

#!/bin/bash

read -p "Enter First Name: " fname

read -p "Enter Middle Name: " mname

read -p "Enter Last Name: " lname

echo "Hello Swayamsiddha A Mane Deshmukh, 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.

#!/bin/bash

ls -l | awk 'NR>1 {print $9, $5}' | sort -nk2

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

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

echo "File with Maximum Size: $max\_file"

echo "File with 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)

#!/bin/bash

day=$(date +%u)

if [[ $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.

#!/bin/bash

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

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

echo "Welcome to the HP Health Club!"

else

echo "Sorry, your weight is outside the acceptable range."

fi

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

#!/bin/bash

hour=$(date +%H)

if [[ $hour -lt 12 ]]; then

echo "Good Morning!"

elif [[ $hour -lt 18 ]]; then

echo "Good Afternoon!"

else

echo "Good Evening!"

fi

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.

#!/bin/bash

file="students.txt"

# Accept roll number

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

# Search for the roll number

record=$(grep "^$roll\_no:" "$file")

if [ -z "$record" ]; then

echo "Roll No Not Found"

else

echo "Current Record: $record"

read -p "Enter New Name: " name

read -p "Enter New Marks for Hindi: " marks\_hindi

read -p "Enter New Marks for Maths: " marks\_maths

read -p "Enter New Marks for Physics: " marks\_physics

# Create new record

new\_record="$roll\_no:$name:$marks\_hindi:$marks\_maths:$marks\_physics"

sed -i "s|^$record|$new\_record|" "$file"

echo "Record Updated Successfully"

fi

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

#!/bin/bash

file="students.txt"

# Check if roll number is provided

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 "Current Record: $record"

read -p "Enter New Name: " name

read -p "Enter New Marks for Hindi: " marks\_hindi

read -p "Enter New Marks for Maths: " marks\_maths

read -p "Enter New Marks for Physics: " marks\_physics"

new\_record="$roll\_no:$name:$marks\_hindi:$marks\_maths:$marks\_physics"

sed -i "s|^$record|$new\_record|" "$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.

#!/bin/bash

file="students.txt"

# Check if roll number is provided

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

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

sed -i "/^$roll\_no:/d" "$file"

echo "Record Deleted Successfully"

else

echo "Delete Operation Cancelled"

fi

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.

#!/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 a regular file."

elif [ -d "$file" ]; then

echo "$file is a directory."

else

echo "$file is of an 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

#!/bin/bash

file="students.txt"

log="log1"

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

read -p "Enter Name: " name

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

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

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

record="$roll\_no:$name:$marks\_hindi:$marks\_maths:$marks\_physics"

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

echo "$record - roll number exists" >> "$log"

echo "Roll number exists. Logged to $log."

elif [[ $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 "$record - marks out of range" >> "$log"

echo "Marks out of range. Logged to $log."

else

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 50 ]; then

grade="C"

else

grade="D"

fi

echo "$record:$total:$percentage:$grade" >> "$file"

echo "Record Added Successfully with Total=$total, Percentage=$percentage%, Grade=$grade."

fi