

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

```
*
**
***
****

#!/bin/bash

for i in {1..4}
do
    for j in $(seq 1 $i)
    do
        echo -n "*"
    done
    echo ""
done
```

2 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

echo " Enter your first name:"
read fname
echo "Enter your middle name:"
read mname
echo "Enter your last name:"
read lname
echo "Hello, $fname $mname $lname! Welcome to Linux!"
```

Output:

Enter your middle name:

Ramesh

Enter your last name:

Shewale

Hello, Shravani Ramesh Shewale! Welcome to Linux!

- 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.**

```
#!/bin/bash
```

```
# List all files in the current directory
```

```
echo "Files in the current directory:"
```

```
ls -l
```

```
# file with the maximum size
```

```
max_file=$(ls -S | head -n 1)
```

```
max_size=$(stat -c%s "$max_file")
```

```
# 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)"
```

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

(Note: Working day Mon-Fri)

```
#!/bin/bash
```

```
##current day of week in number (1 - mon; 7-sun)
```

```
day=$(date +%u)
```

```
#check if between 1 to 5 (mon to fri)
```

```
if [ "$day" -ge 1 ] && [ "$day" -le 5 ]; then
```

```
    echo "It is a working day"
```

```
else
```

```
    echo "It is not a working day"
```

fi

Output:

```
[admin@hostname01 ~]$ ./day.sh
```

It is a working day

- 5 Write a script that accepts a member into HP health club, if the weight of the person is within the range of 30-250 Kgs.**

```
#!/bin/bash
```

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

```
read weight
```

```
#compare weights
```

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

```
then
```

```
    echo "Welcome to the HP Health Club!!"
```

```
else
```

```
    echo "Sorry, your weight is not in given weight range for membership"
```

```
fi
```

OUTPUT

```
Enter your weight in kg:
```

```
60
```

```
Welcome to the HP Health Club!!
```

```
[admin@hostname01 ~]$ ./health_club.sh
```

```
Enter your weight in kg:
```

```
270
```

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

```
#!/bin/bash
```

```
# Get time in 24hr formate
```

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

```

if [ "$hour" -ge 5 ] && [ "$hour" -lt 12 ]; then
    greeting="Good morning!"
elif [ "$hour" -ge 12 ] && [ "$hour" -lt 17 ]; then
    greeting="Good afternoon!"
elif [ "$hour" -ge 17 ] && [ "$hour" -lt 21 ]; then
    greeting="Good evening!"
else
    greeting="Good night!"
fi

```

```

# Display the greeting
echo $greeting

```

OUTPUT:

```

[admin@hostname01 ~]$ ./greet.sh
Good Evening!

```

- 7 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
echo "Enter the roll number:"
read rollno

# check roll no exist
record=$(grep "^$rollno:" students.txt)

if [ -z "$record" ]; then
    echo "Roll No Not Found"
else
    # Extract the current details
    old_name=$(echo "$record" | cut -d ':' -f 2)

```

```
old_marks1=$(echo "$record" | cut -d ':' -f 3)
old_marks2=$(echo "$record" | cut -d ':' -f 4)
old_marks3=$(echo "$record" | cut -d ':' -f 5)
```

```
# enter new details
```

```
echo "Current Name: $old_name"
```

```
echo "Enter new name:"
```

```
read new_name
```

```
echo "Current Marks in Subject 1: $old_marks1"
```

```
echo "Enter new marks for Subject 1:"
```

```
read new_marks1
```

```
echo "Current Marks in Subject 2: $old_marks2"
```

```
echo "Enter new marks for Subject 2:"
```

```
read new_marks2
```

```
echo "Current Marks in Subject 3: $old_marks3"
```

```
echo "Enter 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/^\$record$/\$new_record/" students.txt
```

```
echo "Record updated "
```

```
fi
```

```
[admin@hostname01 ~]$ ./record.sh
Enter the roll number:
102
Current Name: Shravani Shewale
Enter New Name:
nive
Current Marks in Subjects1: 90
Enter new marks for Sunject 1:
89
Current Marks in Subjects2: 78
Enter new marks for Sunject 2:
96
Current Marks in Subjects3: 84
Enter new marks for Sunject 3:
77
Record Updated
```

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

```
#!/bin/bash
# student records file
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
record=$(grep "^$rollno:" "$data_file")
```

```

if [ -z "$record" ]; then
    echo "Roll No Not Found"
else
    # Display the current record
    echo "Current record: $record"

    # 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 !!"
fi

```

- 9 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 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

```

10

11 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
# 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>
```

- 12 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

- a If the roll number already exists, then store the record and the following message “roll number exists” in a log file “log1”.
- b 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”
- c If the data is valid, the calculate total, percentage, grade and display on the terminal

```
#!/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"
```