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

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

Ans.
```
[admin@hostname01 ~]$ #!/bin/bash

# Number of rows in the triangle
rows=4

# Loop through rows
for ((i = 1; i <= rows; i++))
do
   # Print stars for each row
   for ((j = 1; j <= i; j++))
   do
      echo -n "*"
   done
   echo # Move to the next line
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.

Ans.
```
[admin@hostname01 ~]$ #!/bin/bash
# Prompt the user for their first name, middle name, and last name
read -p "Enter your first name: " fname
read -p "Enter your middle name: " mname
read -p "Enter your last name: " lname

# Concatenate the full name
fullname="$fname $mname $lname"
```

```
# Greet the user
echo "Hello, $fullname! Welcome!"
```

Enter your first name: Annelies

Enter your middle name: Marie

Enter your last name: Frank

Hello, Annelies Marie Frank! 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. [admin@hostname01 ~]$ #!/bin/bash

```
# Display all files in the current directory with their sizes
echo "Files in the current directory:"
ls -lh | awk '{print $9, $5}' | tail -n +2

# Check if there are any files
if [[ $(ls -l | wc -l) -le 1 ]]; then
    echo "No files in the current directory."
    exit 1
fi

# Find the file with the maximum size
max_file=$(ls -S | head -1)
max_size=$(ls -lS | awk 'NR==2 {print $5}')

# Find the file with the minimum size
min_file=$(ls -Sr | head -1)
min_size=$(ls -lSr | awk 'NR==2 {print $5}')

# Display results
echo -e "\nFile with the maximum size: $max_file ($max_size bytes)"
echo "File with the minimum size: $min_file ($min_size bytes)"
```

Files in the current directory:

add.c 0

chap1 0

Desktop 6

Documents 6

Downloads 50

Music 6

newdir 6

Pictures 6

Public 6

Templates 6

Videos 6

File with the maximum size: Downloads (50 bytes)

File with the minimum size: chap1 (0 bytes)

4 Write a script which when executed checks out whether it is a working day or not? (Note: Working day Mon-Fri)

admin@hostname01 ~]$ #!/bin/bash

Get the current day of the week

day=$(date +%u)

Check if it's a working day (Monday to Friday)

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

  echo "Today is a working day."

else

  echo "Today is not a working day."

fi

Today is a working day.

5 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. [admin@hostname01 ~]$ nano hp.sh

!/bin/bash

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

if ["$weight" -lt 30] || ["$weight" -gt 250]; then

```
echo "Sorry, your weight is outside the acceptable range (30-250 kg). You >
else
echo "Welcome to HP Health Club! You have been successfully accepted."
fi
```

```
[admin@hostname01 ~]$ chmod +x hp.sh
[admin@hostname01 ~]$ ./hp.sh
Enter your weight in kg: 59
Welcome to HP Health Club! You have been successfully accepted.
```

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

Ans. [admin@hostname01 ~]$ #!/bin/bash

```
# Get the current hour
hour=$(date +%H)

# Greet based on the hour of the day
if [ "$hour" -lt 12 ]; then
   echo "Good Morning!"
elif [ "$hour" -lt 18 ]; then
   echo "Good Afternoon!"
else
   echo "Good Evening!"
fi
```

```
Good Morning!
```

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.

Ans. [admin@hostname01 ~]$ nano modfify_stu.sh

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

```
file="student_records.txt"
read -p "Enter the roll number to search: " rollno
      record=$(grep "^$rollno:" "$file")
if [ -n "$record" ]; then
  echo "Record found: $record"
  IFS=":" read -r roll name marks1 marks2 marks3 <<< "$record"

  # Allow user to modify name and marks for modification
  read -p "Enter new name (current: $name): " new_name
  read -p "Enter new mark for subject 1 (current: $marks1): " new_marks1
  read -p "Enter new mark for subject 2 (current: $marks2): " new_marks2
  read -p "Enter new mark for subject 3 (current: $marks3): " new_marks3

  sed -i
"s/^$rollno:$name:$marks1:$marks2:$marks3$/$rollno:$new_name:$new_marks1:$new_marks2:$new_marks3
/" "$file"

  echo "Record updated successfully!"
else
  # If roll number is not found
  echo "Roll No Not Found"
fi
```

```
[admin@hostname01 ~]$ ./modfify_stu.sh
Enter the roll number to search: 7
Record found: 6:Ritul:85:90:95
Enter new name (current: Ritul): Isha
Enter new mark for subject 1 (current: 80): 90
Enter new mark for subject 2 (current: 90): 92
Enter new mark for subject 3 (current: 92): 95
Record updated successfully!
[admin@hostname01 ~]$ cat studentrecord.txt
6:Ritul:80:90:92
7:Isha:90:92:95
8:Rani:75:80:85
```

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

Ans. [admin@hostname01 ~]$ nano modfify_stu.sh

```
# To Accept roll number
read -p "Enter the roll number to search: " rollno
```

[admin@hostname01 ~]$ ./modfify_stu.sh

Enter the roll number to search: 7

Record found: 7:Isha:90:92:95

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.

Ans.[admin@hostname01 ~]$ nano del_stu.sh

```
#!/bin/bash
file="studentrecord.txt"
if [ -z "$1" ]; then
 read -p "Enter the roll number to search: " rollno
else
  # Use the command line argument for roll number
  rollno=$1
fi
record=$(grep "^$rollno:" "$file")
if [ -n "$record" ]; then
  echo "Record found: $record"
  read -p "want to delete record type y : " confirm



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

    sed -i "/^$rollno:/d" "$file"
    echo "Record deleted successfully!"
  else
    echo "Deletion aborted."
  fi
```

```
else

  echo "Roll No Not Found"
fi
[admin@hostname01 ~]$ chmod +x del_stu.sh
[admin@hostname01 ~]$ ./del_stu.sh
Enter the roll number to search: 8
Record found: 8:Rani:75:80:85
want to delete record type y : y
Record deleted successfully!
```

10 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.
```
[admin@hostname01 ~]$ nano filereport.sh
  if [ "$#" -ne 1 ]; then
      echo "Error: Please provide exactly one argument."
      echo "Usage: $0 <file_path>"
      exit 1
    fi
    file_path=$1
    if [ -e "$file_path" ]; then
      file_type=$(file "$file_path")
      echo "$file_type"
    else
      echo "Error: $file_path does not exist."
      exit 1
    fi
    [admin@hostname01 ~]$ chmod +x filereport.sh
    [admin@hostname01 ~]$ ./filereport.sh networrk.txt
    networrk.txt: ASCII text, with very long lines
    [admin@hostname01 ~]$ ./filereport.sh errorfile.txt
    Error: errorfile.txt does not exist.
    [admin@hostname01 ~]$ ./filereport.sh styles
    styles: directory
```

11 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
Ans. #!/bin/bash

```
student_file="student"
log_file="log1"

read -p "Enter Roll No: " roll_no
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


if grep -q "^$roll_no:" "$student_file"; then
  echo "Roll number $roll_no exists."
  echo "roll number exists" >> "$log_file"
  exit 1
fi


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 "Marks out of range."
  echo "marks out of range" >> "$log_file"
  exit 1
fi

echo "$roll_no:$name:$marks_hindi:$marks_maths:$marks_physics" >> "$student_file"

total=$((marks_hindi + marks_maths + marks_physics))
percentage=$((total / 3))

echo "Total Marks: $total"
echo "Percentage: $percentage%"
```

a. If the roll number already exists, then store the record and the following message “roll number exists” in a log file “log1”.

Ans. [admin@hostname01 ~]$ ./studentrec.sh

Enter Roll No: 8

Enter Name: Rani

Roll number 8 exists.

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”

Ans. [admin@hostname01 ~]$ ./studentrec.sh

Enter Roll No: 10

Enter Name: Gayatri

Enter Marks in Hindi: 95

Enter Marks in Maths: 88

Enter Marks in Physics: 100

Marks out of range.

c. If the data is valid, the calculate total, percentage, grade and display on the terminal

Ans. [admin@hostname01 ~]$ ./studentrec.sh
Enter Roll No: 12
Enter Name: Rashmi
Enter Marks in Hindi: 92
Enter Marks in Maths: 90
Enter Marks in Physics: 85
Total Marks: 267
Percentage: 89%

* Function to validate email address

[admin@hostname01 ~]$ nano gmail.sh

```
#!/bin/bash

read -p "Enter your email address: " email

email_patt="^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.(com|in)$"

if [[ $email =~ $email_patt ]]; then
echo "The email address is valid."
else
echo "Invalid email address."
fi



# Prompt user for email address
echo -n "Enter your email address: "
read email

# Validate the email address
if validate_email "$email"; then
    echo "The email address '$email' is valid."
```

```
else
   echo "The email address '$email' is invalid."
Fi
```

```
[admin@hostname01 ~]$ chmod +x gmail.sh
[admin@hostname01 ~]$ ./gmail.sh
Enter your email address: tweetlady@gmail.com
The email address is valid.
[admin@hostname01 ~]$ ./gmail.sh
Enter your email address: tweety@aec.ac.in
Invalid email address.
```

*Function for validating mobile number

```
#!/bin/bash

read -p "Enter your mobile number: " mobile_number

mobile_patt="^\+91(89|90)[0-9]{8}$"

if [[ $mobile_number =~ $mobile_patt ]]; then
  echo "The mobile number is valid."
else
  echo "Invalid mobile number. Ensure it starts with +91 and begins with 89 or 90."
fi
```

```
[admin@hostname01 ~]$ nano mobilechk.sh
[admin@hostname01 ~]$ chmod +x mobilechk.sh
[admin@hostname01 ~]$ ./mobilechk.sh
Enter your mobile number: 8867453421
Invalid mobile number. Ensure it starts with +91 and begins with 89 or 90.
[admin@hostname01 ~]$ ./mobilechk.sh
Enter your mobile number: +918976231564
The mobile number is valid.
[admin@hostname01 ~]$
```