1. If -else batch scripting

Create a file vi if-script.bash and then write:

```
#!/bin/bash
count=100
if [ $count -eq 100 ]
    then
    echo count is 100
else
    echo count is not 100
fi
~
```

Change the permissions to chmod755 vi if-script.bash and then execute it by bash if-script.bash or ./if-script.bash:

```
[root@ip-172-31-42-219 scripts]# bash if-script.bash
count is 100
```

Another example for if -else

Create a file with vi

```
#!/bin/bash
if [ -e pwd ]
then
        echo "file exist"
else
        echo "file doesnt exist"
fi
~
```

Change the permissions to chmod755 vi file.bash and then execute it by bash file.bash or ./bash file.bash:

```
[root@ip-172-31-42-219 scripts]# bash file.bash
file doesnt exist
```

- Case statements:

  Create a file vi case.bash

```
#!/bin/bash
echo
echo Please chose one of the options below
echo
echo 'a = Display Date and Time'
echo 'b = List file and directories'
echo 'c = List users logged in'
echo 'd = Check System uptime'
echo
        read choices
        case $choices in
a) date;;
b) ls;;
c) who;;
d) uptime;;
*) echo Invalid choice - Bye
esac
```

Change the permissions to chmod755 case.bash and then execute it by bash case.bash or ./bash case.bash:

```
[root@ip-172-31-42-219 scripts]# vi case.bash
[root@ip-172-31-42-219 scripts]# chmod 755 case.bash
[root@ip-172-31-42-219 scripts]# bash case.bash

Please chose one of the options below

a = Display Date and Time
b = List file and directories
c = List users logged in
d = Check System uptime

a
Fri Sep  5 10:12:22 UTC 2025
```

If any invalid option given to that it shows as invalid:

```
[root@ip-172-31-42-219 scripts]# bash case.bash

Please chose one of the options below

a = Display Date and Time
b = List file and directories
c = List users logged in
d = Check System uptime

f
Invalid choice - Bye
[root@ip-172-31-42-219 scripts]#
```

- For loop :

  Create a file vi for.bash

```
#!/bin bash

for i in 1 2 3 4 5
do
        echo "welcome $i times"
done
~
```

Change the permissions to chmod755 for.bash and then execute it by bash for.bash or ./bash for.bash:

```
[root@ip-172-31-42-219 scripts]# vi for.bash
[root@ip-172-31-42-219 scripts]# chmod 755 for.bash
[root@ip-172-31-42-219 scripts]# bash for.bash
welcome 1 times
welcome 2 times
welcome 3 times
welcome 4 times
welcome 5 times
[root@ip-172-31-42-219 scripts]#
```

Another example for for loop in that file only we edit vi for.bash

```
[root@ip-172-31-42-219 scripts]# vi for.bash
[root@ip-172-31-42-219 scripts]# ./for.bash
see mujaheed eat
see mujaheed run
see mujaheed jump
see mujaheed walk
[root@ip-172-31-42-219 scripts]#
```

- Do while:

  Create a file vi dowhile.bash

```
#!/bin/bash

count=0
num=10
while [ $count -lt 10 ]
do
        echo
        echo $num seconds left to stop this process $1
        echo
        sleep 1

        num='expr $num - 1'
        count='expr $count + 1'
done
echo
echo $1 process is stopped!!!
echo
```

Change the permissions to chmod755 dowhile.bash and then execute it by bash dowhile.bash or ./bash dowhile.bash:

```
[root@ip-172-31-42-219 scripts]# bash dowhile.bash
10 seconds left to stop this process

9 seconds left to stop this process

8 seconds left to stop this process

7 seconds left to stop this process

6 seconds left to stop this process

5 seconds left to stop this process

4 seconds left to stop this process

3 seconds left to stop this process

2 seconds left to stop this process

1 seconds left to stop this process

process is stopped!!!
```

Exit status:

To check the exit status

0-successful

1=minor problem

2=serious problem

3-255=everything else

```
[root@ip-172-31-42-219 scripts]# echo $?
0
[root@ip-172-31-42-219 scripts]# pwdi
bash: pwdi: command not found
[root@ip-172-31-42-219 scripts]# echo $?
127
[root@ip-172-31-42-219 scripts]# cd
[root@ip-172-31-42-219 ~]# echo $?
0
[root@ip-172-31-42-219 ~]#
```

- Create a file exitstatus.bash

```
#!/bin/bash

pwd
if [ $? -eq 0 ]
then
        echo file exist
else
        echo file does not exist
fi
~
```

Change the permissions to chmod755 exitstatus.bash and then execute it by bash exitstatus.bash or ./bash exitstatus.bash:

```
[root@ip-172-31-42-219 scripts]# vi exitstatus.bash
[root@ip-172-31-42-219 scripts]# bash exitstatus.bash
/mujju/scripts
file exist
[root@ip-172-31-42-219 scripts]#
```

- Cronjob :

  We need to install cronjob package - yum install cronie

  And then edit it *****/root/test:

```
* * * * * /root/test
```

```
[root@ip-172-31-42-219 ~]# ls
test
[root@ip-172-31-42-219 ~]# date
Fri Sep  5 12:34:58 UTC 2025
```

Bash script tasks :

1. Create a bash script to **check if a directory is available or not**.

Create a file vi if.bash and add:

```
#!/bin/bash

dir=$home/mujju/scripts
if [ -d "$dir" ];
then
        echo "Directory '$dir' exists"
else
        echo "directory '$dir' does not exists"
fi
```

Chmod 755 if.bash and execute it bash if.bash or ./if.bash:

```
[root@ip-172-31-42-219 scripts]# bash if.bash
Directory '/mujju/scripts' exists
```

2. Create a bash script to **create multiple files**.

   Create a file vi for.bash and add:

```
#!/bin/bash

for i in {1..7}
do
        touch $i
done
[root@ip-172-31-42-219 scripts]#
```

Chmod 755 for.bash and execute it bash for.bash or ./for.bash:

```
[root@ip-172-31-42-219 scripts]# bash for.bash
[root@ip-172-31-42-219 scripts]# ls
1  6           c               file.bash        if.bash
2  7           case.bash       for.bash         input.bash
3  a           commands.bash   forloop.bash     variables.bash
4  admin.bash  dowhile.bash    helloworld.bash
5  b           exitstatus.bash if-script.bash
```

3 . Create a bash script to **take a backup of a directory**.

Create a file vi backupdirectory.bash

```
#!/bin/bash
read -p "Enter the directory you want to backup: " src

if [ ! -d "$src" ];
then
        echo "Error: Directory '$src' does not exist."
        exit 1
fi
backup_dir="$HOME/backups"

mkdir -p "$backup_dir"

backup_name="$(basename "$src")-$(date +%Y%m%d_%H%M%s).tar.gz"
tar -czf "$backup_dir/$backup_name" -c "$(dirname "$src")" "$(basename "$src")"
echo "Backup of 'src' created at: $backup_dir/$backup_name"
[root@ip-172-31-42-219 scripts]#
```

Chmod 755 backupdirectory.bash and execute it bash backupdirectory.bash or ./backupdirectory.bash:

```
[root@ip-172-31-42-219 scripts]# bash backupdirectory.bash
Enter the directory you want to backup: /mujju/scripts
tar: Removing leading `/' from member names
tar: Removing leading `/' from hard link targets
tar: scripts: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
Backup of 'src' created at: /root/backups/scripts-20250905_13561757080603.tar.gz
```

4 . Create a bash script to install Nginx on an EC2 server

Execute command vi nginxec2.bash and add script in it

```
#!/bin/bash
sudo yum update -y
sudo yum install nginx -y
sudo systemctl enable nginx
sudo systemctl start nginx
echo "nginx has been installed and started"
~
```

Chmod 755 nginx.bash and execute it bash nginxec2.bash or ./nginxec2.bash:

```
[root@ip-172-31-42-219 scripts]# bash nginx.bash
☑ Nginx is already running.
[root@ip-172-31-42-219 scripts]#
```

5. Create a bash script to install Apache Tomcat on an EC2 server

Create a file named as apache tomcat.bash and add script in it.

```
#!/bin/bash
# Script to install Apache Tomcat on EC2

# Exit on error
set -e

# ------------------------
# Variables
# ------------------------
TOMCAT_VERSION=9.0.93
TOMCAT_USER=tomcat
INSTALL_DIR=/opt/tomcat

# ------------------------
# Update system
# ------------------------
echo "[INFO] Updating system..."
sudo yum update -y || sudo apt-get update

# ------------------------
# Install Java
# ------------------------
echo "[INFO] Installing Java..."
if command -v yum >/dev/null 2>&1; then
  sudo yum install -y java-11-openjdk
```

Change permissions to chmod 755 apache tomcat.bash and execute it bash apache tomcat.bash or ./apache tomcat.bash

```
[root@ip-172-31-42-219 scripts]# bash apache tomcat.bash
[INFO] Updating system...
Last metadata expiration check: 0:20:54 ago on Fri Sep 12 12:32:24 2025.
================================================================================
WARNING:
  A newer release of "Amazon Linux" is available.

  Available Versions:

  Version 2023.8.20250908:
    Run the following command to upgrade to 2023.8.20250908:

      dnf upgrade --releasever=2023.8.20250908

    Release notes:
     https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.8.20250908.html

================================================================================
Dependencies resolved.
Nothing to do.
```

6. Create a bash script to check if the Nginx service is running, if not running then script should start the service.

Create a file named as nginx.bash and add script to it

```
#!/bin/bash

if systemctl is-active --quiet nginx; then
    echo "☑ Nginx is already running."
else
    echo "⚠ Nginx is not running. Starting service..."
    sudo systemctl start nginx
    if systemctl is-active --quiet nginx; then
        echo "☑ Nginx started successfully."
    else
        echo "✖ Failed to start Nginx."
        exit 1
    fi
fi
```

Change permissions to chmod 755 nginx.bash and execute it bash nginx.bash or ./nginx.bash

```
[root@ip-172-31-42-219 scripts]# bash nginx.bash
☑ Nginx is already running.
[root@ip-172-31-42-219 scripts]#
```

7. Create a Bash script for a calculator

Create a file named as calculator.bash and add script to it

```
#!/bin/bash

while true; do
    echo "============================="
    echo "       Simple Calculator     "
    echo "============================="
    echo "1. Addition"
    echo "2. Subtraction"
    echo "3. Multiplication"
    echo "4. Division"
    echo "5. Exit"
    echo "============================="
    read -p "Enter your choice: " choice

    case $choice in
        1)
            read -p "Enter first number: " a
            read -p "Enter second number: " b
            echo "Result = $((a + b))"
            ;;
        2)
            read -p "Enter first number: " a
            read -p "Enter second number: " b
            echo "Result = $((a - b))"
            ;;
"caluclator.bash" 50L, 1304B
```

Change permissions to chmod 755 caluclator.bash and execute it bash caluclator.bash or ./caluclator.bash

```
-bash: ./: is a directory
[root@ip-172-31-42-219 scripts]# ./caluclator.bash
============================
      Simple Calculator
============================
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
============================
Enter your choice: 2
Enter first number: 10
Enter second number: 4
Result = 6

============================
      Simple Calculator
============================
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
```

8. Create a bash script to check if the directory is available or not. Ifnot, then create a directory

Create a file named as directorycreate.bash

```
#!/bin/bash
# Script to check if a directory exists, if not then create it

# Directory name (you can change this)
DIR_NAME="myfolder"

# Check if directory exists
if [ -d "$DIR_NAME" ]; then
    echo "[INFO] Directory '$DIR_NAME' already exists."
else
    echo "[INFO] Directory '$DIR_NAME' does not exist. Creating now..."
    mkdir -p "$DIR_NAME"
    echo "[SUCCESS] Directory '$DIR_NAME' created."
fi
```

Change permissions to chmod 755 directorycreate.bash and execute it bash directorycreate.bash or ./directorycreate.bash

```
[root@ip-172-31-42-219 scripts]# bash directorycreate.bash
[INFO] Directory 'myfolder' already exists.
```

9. Create a bash script to delete last 3 lines.

Create a file with name delete.bash

```
[root@ip-172-31-42-219 scripts]# cat del.bash
#!/bin/bash
echo "enter the name of the file:"
read file

head -n -3 "$file" > temp.txt
mv temp.txt "$file"
echo "last 3 lines deleted."
```

Change permissions to chmod 755 delete.bash and execute it bash delete.bash or ./delete.bash

```
[root@ip-172-31-42-219 scripts]# vi del.bash
[root@ip-172-31-42-219 scripts]# chmod 755 del.bash
[root@ip-172-31-42-219 scripts]# bash del.bash
enter the name of the file:
dollar
head: cannot open 'dollar' for reading: No such file or director
y
last 3 lines deleted.
[root@ip-172-31-42-219 scripts]#
```

10. Create a bash script to monitor CPU. If the usage is more than 80%, then send a notification.

Create a file with name called as vi notification.bash and add the script

```bash
#!/bin/bash
# Script to monitor CPU usage and send a notification if usage > 80%

# Threshold
THRESHOLD=80

# Get CPU usage (average across all cores)
CPU_USAGE=$(top -bn1 | grep "Cpu(s)" | awk '{print 100 - $8}')

# Convert to integer (remove decimals)
CPU_INT=${CPU_USAGE%.*}

echo "[INFO] Current CPU Usage: $CPU_INT%"

# Check if usage > threshold
if [ "$CPU_INT" -gt "$THRESHOLD" ]; then
    echo "[ALERT] CPU usage is above $THRESHOLD% !"

    # Send system notification (Linux desktop)
    if command -v notify-send >/dev/null 2>&1; then
        notify-send "High CPU Alert" "CPU usage is at $CPU_INT%"
    fi

    # Log alert
    echo "$(date): High CPU usage detected ($CPU_INT%)" >> cpu_alert.log
fi
```

Change permissions to chmod 755 notification.bash and execute it bash notification.bash or ./notification.bash

```
[root@ip-172-31-42-219 scripts]# bash notification.bash
[INFO] Current CPU Usage: 3%
[root@ip-172-31-42-219 scripts]#
```

11. Create a bash script to monitor disk space, and if it is more than 80% then send a notification

Create a file named as disk space and add script into it

```bash
#!/bin/bash

THRESHOLD=80
EMAIL="receiver-email@example.com"

# Check all real partitions (ignoring tmpfs, udev, loop devices)
df -hP | awk 'NR>1 && $1 !~ /tmpfs|udev|loop/ {print $5 " " $6}' | while read output; do
    usage=$(echo $output | awk '{print $1}' | tr -d '%')
    partition=$(echo $output | awk '{print $2}')

    echo "Partition: $partition - Usage: ${usage}%"

    if [ "$usage" -gt "$THRESHOLD" ]; then
        echo "⚠ Alert: Disk usage on $partition is ${usage}% (above $THRESHOLD%)"
        echo -e "Subject: High Disk Usage Alert\n\nWarning: Disk usage on $partition is at ${usage}%." | msmtp
$EMAIL
    fi
done
```

Change permissions to chmod 755 diskspace.bash and execute it bash diskspace.bash or ./diskspace.bash

```
[root@ip-172-31-42-219 scripts]# bash diskspace.bash
Partition: / - Usage: 26%
Partition: /boot/efi - Usage: 13%
```

12. Bash script to monitor memory and if it is more than 80% then send email notification.

Create a file with the name memory.bash and add script to it

```
#!/bin/bash

THRESHOLD=80
EMAIL="receiver-email@example.com"
INTERVAL=10   # seconds between checks

while true; do
    # Get memory usage percentage
    MEM_USAGE=$(free | awk '/Mem/ {printf("%.0f"), $3/$2 * 100}')

    echo "Current Memory Usage: $MEM_USAGE%"

    if [ "$MEM_USAGE" -gt "$THRESHOLD" ]; then
        echo "Memory usage is above $THRESHOLD%. Sending email..."
        echo -e "Subject: High Memory Usage Alert\n\nWarning: Memory usage is at $MEM_USAGE%" | msmtp $EMAIL
    fi

    sleep "$INTERVAL"
done
```

Change permissions to chmod 755 memory.bash and execute it bash memory.bash or ./memory.bash

```
[root@ip-172-31-42-219 scripts]# bash memory.bash
Current Memory Usage: 35%
```