1)Write a Script that will monitor the ram usage. If it is greater than 80% it should print that ram usage is high. If less than 80% it should say ram utilisation is normal.

```
root@ubuntu-VirtualBox:/home/ubuntu# cat /proc/meminfo
MemTotal:        4001004 kB
MemFree:         2331280 kB
MemAvailable:    3026068 kB

root@ubuntu-VirtualBox:/home/ubuntu# vi ram.sh

#!/bin/bash

while true; do
  memt=$(grep MemTotal /proc/meminfo | awk '{print $2}')
  mema=$(grep MemAvailable /proc/meminfo |awk '{print $2}')
  #memp=$( (memt - mema ) * 100)
  memu=$(( ( ( (memt - mema ) * 100 ) / memt ) ))

  if [ "$memu" -gt 80 ]; then
    echo "RAM usage is high : ${memu}%"
  else
    echo "RAM usage normal : ${memu}%"
  fi

  sleep 5
done

root@ubuntu-VirtualBox:/home/ubuntu# rm .ram.sh.swp
root@ubuntu-VirtualBox:/home/ubuntu# vi ram.sh
root@ubuntu-VirtualBox:/home/ubuntu# chmod +x ram.sh
root@ubuntu-VirtualBox:/home/ubuntu# bash ram.sh

RAM usage normal : 24%

root@ubuntu-VirtualBox:/home/ubuntu# free -m -h
               total        used        free      shared  buff/cache   available
Mem:           3.8Gi       702Mi       2.2Gi        28Mi       928Mi       2.9Gi

root@ubuntu-VirtualBox:/home/ubuntu# free | grep Mem | awk '{print $3/$2 *100.0}   '
17.9823
```

2)write a script which will get the process name as the input. If the process is running it should print ID and current memory usage of the particular process in human readable format . if the process is not in running state print process is not running.

```
root@ubuntu-VirtualBox:/home/ubuntu# vi pros.sh

#!/bin/bash
```

```
read -p "enter process name: " pname

pid=$(pgrep -o "$pname")

if [ -n "$pid" ]; then
  mem=$(ps -o rss= -p "$pid")
  echo "ProcessID: $pid"
  echo "Mem usage: ${mem}K"
else
  echo "process $pname is not running"
fi
```

```
root@ubuntu-VirtualBox:/home/ubuntu# chmod +x pros.sh
root@ubuntu-VirtualBox:/home/ubuntu# bash pros.sh
enter process name: mike
process mike is not running
root@ubuntu-VirtualBox:/home/ubuntu# bash pros.sh
enter process name: bash
ProcessID: 1907
Mem usage:  5120K
```

3)write a script that will print the cpu load in last 1 min, 5min, 15min, current cpu utilisation , disk usage and memory utilisation in percentage

```
root@ubuntu-VirtualBox:/home/ubuntu# vi cpu.sh
root@ubuntu-VirtualBox:/home/ubuntu# chmod +x cpu.sh
root@ubuntu-VirtualBox:/home/ubuntu# bash cpu.sh
```

```
#!/bin/bash


load_averages=$(uptime | awk -F 'load average: ' '{print $2}')

load_1min=$(echo $load_averages | cut -d, -f1)

load_5min=$(echo $load_averages | cut -d, -f2)

load_15min=$(echo $load_averages | cut -d, -f3)


cpu_utilization=$(top -bn1 | grep "Cpu(s)" | awk '{print 100 - $8"%"}')

disk_usage=$(df -h --total | grep 'total' | awk '{print $5}')

memory_utilization=$(free | grep Mem | awk '{print $3/$2 * 100.0}')
```

```
echo "CPU Load Averages:"
echo "Last 1 minute: $load_1min"
echo "Last 5 minutes: $load_5min"
echo "Last 15 minutes: $load_15min"
echo "Current CPU Utilization: $cpu_utilization"

echo "Disk Usage: $disk_usage"
echo "Memory Utilization: $(printf  $memory_utilization)%"
```

```
CPU Load Averages:
Last 1 minute: 0.00
Last 5 minutes:  0.00
Last 15 minutes:  0.00
Current CPU Utilization: 3.1%
Disk Usage: 35%
Memory Utilization: 17.5471%

root@ubuntu-VirtualBox:/home/ubuntu# uptime
 18:51:50 up 23:53,  3 users,  load average: 0.00, 0.00, 0.00
root@ubuntu-VirtualBox:/home/ubuntu# uptime -p
up 23 hours, 53 minutes
```

4)write a script that should get username as input and it should create user if doesnot exist check whether password is set for particular user name.

```
root@ubuntu-VirtualBox:/home/ubuntu# vi user.sh

#!/bin/bash

read -p "enter username: " username

if id "$username" &>/dev/null; then
  echo "user exists"

  if sudo grep -q "^username:" /etc/shadow && sudo awk -F: -v user="$username" '$1==user
&& $2!=$3{exit 1}' /etc/shadow; then
    echo "passwd is set"
  else
    echo "not set"
  fi

else
  sudo useradd "$username"
  echo "usr created"
fi
```

~

```
root@ubuntu-VirtualBox:/home/ubuntu# chmod +x user.sh
root@ubuntu-VirtualBox:/home/ubuntu# bash user.sh
enter username: mike
usr created

root@ubuntu-VirtualBox:/home/ubuntu# bash user.sh
enter username: ubuntu
user exists
passwd is set
```

5)write a script that can go to directory and it should find specific file. If size is greater than 100mb it should tar zip the file, delete the file and create a file with same name

```
#!/bin/bash

read -p "Enter filename: " FILENAME
cd "/home/ubuntu" ||  exit 1
if [[ -f "$FILENAME" ]]; then
  if [[ $(stat -c%s "$FILENAME") -gt 104857600 ]]; then
    tar -cvzf "${FILENAME}.tar.gz" "$FILENAME" && rm "$FILENAME" && touch
"$FILENAME"
    echo "File is tared, deleted, and recreated."
  else
    echo "less than 100mb"
  fi
else
  echo "File $FILENAME does not exist."
fi

root@ubuntu-VirtualBox:/home/ubuntu# touch example.txt
root@ubuntu-VirtualBox:/home/ubuntu# echo "This is a test file." > example.txt
root@ubuntu-VirtualBox:/home/ubuntu# bash tar.sh

Enter filename: example.txt
less than 100mb

root@ubuntu-VirtualBox:/home/ubuntu# truncate -s 150M largefile.txt
root@ubuntu-VirtualBox:/home/ubuntu# bash tar.sh
Enter filename: largefile.txt
largefile.txt
File is tared, deleted, and recreated.
```

POWERSHELL

6))write a script which will get the process name as the input. If the process is running it should print ID and current memory usage of the particular process in human readable format . if the process is not in running state print process is not running.

```
$processName = Read-Host "Enter the process name"
$process = Get-Process -Name $processName -ErrorAction SilentlyContinue

if ($process) {
    foreach ($p in $process) {
        $processid = $p.Id
        $memUsage = $p.WorkingSet64 / 1MB
        Write-Output "Process ID: $processid"
        Write-Output "Memory Usage:  $memUsage"
    }
} else {
    Write-Output "Process '$processName' is not running."
}
```

```
Enter the process name: msdtc
Process ID: 5168
Memory Usage:  10.48828125
```

7)the powershell should identify the files older than one day and move them to a backup folder

```
$sourceDir = "C:\Downloads"

$backupDir = "$sourceDir\backup"


if (-not (Test-Path $backupDir)) { New-Item -Path $backupDir -ItemType Directory }


#$currentDate = Get-Date


Get-ChildItem -Path $sourceDir -File | Where-Object { $_.LastWriteTime -lt
(Get-Date).AddDays(-1) } | ForEach-Object {
    Move-Item -Path $_.FullName -Destination $backupDir
    Write-Output "Moved file: $($_.FullName)"
}

if (-not (Get-ChildItem -Path $backupDir -File)) { Write-Output "No files older than one day
were found." }
```

```
    Directory: C:\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         8/7/2024   9:04 AM                backup
No files older than one day were found.
```