

Name= shaik Raakin
Class =3rd sem
Branch = cybersecurity
usn=ENG24CY0192
Roll no:64
section:A
Assignment : 8

11Q)What is a user-defined function in shell scripting? Explain with an example.

sol)

A user-defined function is a reusable block of code with a name, which can be executed anywhere in the script by calling its name. Functions improve modularity and code reuse.

Example:

```
#!/bin/bash
greet() {
    echo "Hello, $1!"
}
greet "Alice"
```

12Q)Write a bash script with a function that multiply two integer numbers.

sol)

The bash script for this function is:

```
#!/bin/bash
multiply(){
    multiply=$((num1*num2))
    Echo "$multiply"
}
num1=10
num2=20
Multiply
```

13Q)Explain how arrays (1D, 2D, and 3D) are declared in bash scripting.

sol)

1D array:

```
array=("apple" "banana" "mango")
```

2D Array (simulate with arrays of arrays, not natively supported):

```
arr1=(1 2)
```

```
arr2=(3 4)
```

```
matrix=( [0]=arr1 [1]=arr2 )
```

3D Array: Bash doesn't support true multidimensional arrays; simulate with nested associative arrays or structured naming.

14Q)Write a shell script to display elements of an array

sol)

The shell script for this is:

```
#!/bin/bash
fruits=("apple" "banana" "mango")
for fruit in "${fruits[@]}";do
    echo "$fruit"
done
```

15Q)What is the purpose of cron in Linux?

sol)

Cron is a system utility in Linux that is very powerful and important. It provides the ability for users and administrators to schedule the automatic execution of scripts, commands, or programs at a desired date and/or time, without any user interaction. Cron is always running in the background as a process called the cron daemon (or crond) that continually checks for scheduled tasks, also referred to as "cron jobs," and executes them based upon scheduled time criteria described in specific configuration files called "crontabs."

16Q)Write a cron job to run a backup script every day at midnight.

sol)

Open the cron table (`crontab -e`) and add:

```
0 0 * * * /path/to/backup.sh
```

17Q)How do you schedule a one-time job using at command?

sol)

First, ensure the `at` package is installed.

Then:

```
echo "sh /path/to/script.sh" | at 10:00
```

This schedules `script.sh` to run at 10 AM.

18Q)Write a script to display disk usage using df and du.

sol)

The shell script for this will be :

```
#!/bin/bash
echo "Disk free (df):"
df -h
echo "Disk usage (du) for home directory:"
du -sh ~
```

Output:

```
raakin@DESKTOP-0FFMSPJ:~/uni_linx$ ./trial.sh
Disk free (df):
Filesystem      Size  Used Avail Use% Mounted on
none            3.9G   0  3.9G  0% /usr/lib/modules/6.6.87.2-microsoft-standard-WSL2
none            3.9G  4.0K  3.9G  1% /mnt/wsl
drivers         238G  214G  25G  90% /usr/lib/wsl/drivers
/dev/sdd        1007G 2.6G  954G  1% /
none            3.9G 112K  3.9G  1% /mnt/wslg
none            3.9G   0  3.9G  0% /usr/lib/wsl/lib
rootfs          3.9G  2.7M  3.9G  1% /init
none            3.9G 808K  3.9G  1% /run
none            3.9G   0  3.9G  0% /run/lock
none            3.9G   0  3.9G  0% /run/shm
none            3.9G  76K  3.9G  1% /mnt/wslg/versions.txt
none            3.9G  76K  3.9G  1% /mnt/wslg/doc
C:\             238G  214G  25G  90% /mnt/c
D:\             931G 349G  583G  38% /mnt/d
snapfuse        51M   51M   0 100% /snap/snapd/25202
snapfuse        74M   74M   0 100% /snap/core22/2133
snapfuse        4.5M  4.5M   0 100% /snap/tree/54
tmpfs           785M  20K  785M  1% /run/user/1000
Disk usage (du) for home directory:
232K  /home/raakin
```

19Q). How can you log the output of a script using the tee command?

sol)

Add `| tee logfile.txt` to any command or script to both display and log output.

Example:

```
echo "Processing data..." | tee process.log
```

20Q). Explain with an example how shell scripting can automate system administration tasks.

sol)

Shell scripting is a versatile and significant tool that allows system administrators to automate a variety of simple or very complex activities on a Linux system. By creating scripts (plain text files listed in a particular order), system administrators can automate normally redundant administrative to-do's done manually-to perform administrative activities more accurately, reducing administrative and human error by doing an activity the same way every time.

An admin can automate certain user account changes, system backups, log rotation, downloads/uploads of software, cleanup of temp files, monitoring of available space, running network diagnostics, reporting scheduled activity, etc. Scripts automate time-consuming tasks while providing reliability: scheduled scripts can be run in the background that maintain service policy without the risk of human reminder/memory.

Example:

```
#!/bin/bash
sudo apt update && sudo apt upgrade -y
sudo apt autoremove -y
echo "System updated and cleaned."
```