LINUX PROGRAMMING

ASSIGNMENT-7

NAME: V S SAIDATTA   USN: ENG24CY0175   ROLLNO:28

1. What is a bash shell script? Give one example.

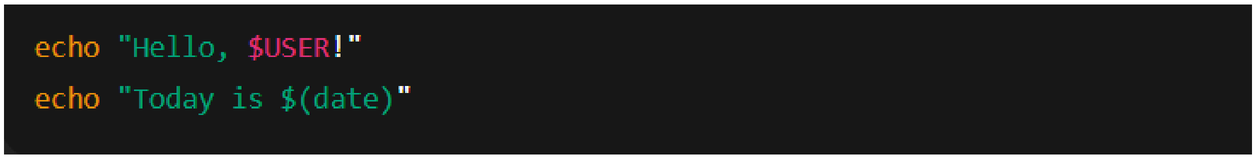
A.

A Bash shell script is a simple text file with the series of commands to be run with the Bash shell (Bourne Again SHell). Bash is a well-known command language and shell for Unix-type operating systems, such as Linux and macOS.

Shell scripts are often used to:

- Automate repetitive tasks.
- Control system operations.
- Sequentially execute multiple commands.
- Work with files and processes.

These are commonly saved with .sh extensions but this is not required.



```
echo "Hello, $USER!"
echo "Today is $(date)"
```

2. Write a simple shell script to print "Hello World".

A.

```
#!/usr/bin/env bash

set -euo pipefail

echo "Hello World"
```

3. What is the purpose of comments (#) in a shell script?

A.

In a shell script, comments are denoted by the # symbol and serve the following purposes:

1. Documentation.
2. Temporarily Disabling code.
3. Improving readability.

Bash ignores everything after # on a line, so comments do not affect runtime.

4. How do you declare variables (int, float, double, string, Boolean, and char in a shell script?

A.

In shell scripts variables are not strictly typed as in C, Java, or Python.All variables are strings, and there are no built-in types such as int, float, double, char, or Boolean.

STRING:

```
name = “Datta”
echo “Name is $name”
```

INTEGER:

```
number = 42
Echo $number
```

FLOAT/DOUBLE:

```
float1=3.5
float2=2.1
result=$(echo “$float1+$float2” | bc)
echo $result
```

BOOLEN:

```
flag=true
If [“$flag” = true]; then
echo “flag is true”
fi
```

CHARACTER:

```
char= ‘A’
```

```
echo $char
```

5. Write a shell script to display the current date and time of the system.
A.

```
#!/usr/bin/env bash

set -euo pipefail

date "+%Y-%m-%d %H:%M:%S %Z"
```

6. Explain the difference between a constant and a variable in bash script.

A.

VARIABLE:

- A variable is an assigned name for a memory location that can hold and change data values while the script is running.
- Declared using the normal assignment operator ‘=’.
- Its value **can be modified** later in the script.
- Syntex: name = “Saidatta”
  echo “name: $name”

CONSTANT:

- A constant is a value that is fixed (the value cannot change once assigned).
- Declared using the readonly or declare -r command.
- Its value **cannot be modified** once defined.
- Syntex: readonly pi=3.14
  echo “PI value: $pi”

7. Write a shell script to read two integer number from the user and compute the sum of both the number.

A.
With robust validation:

```
#!/usr/bin/env bash

set -euo pipefail

read -r -p "Enter first integer: " a
read -r -p "Enter second integer: " b

int_re='^-?[0-9]+$'

if [[ ! "$a" =~ $int_re || ! "$b" =~ $int_re ]]; then

  echo "Error: both inputs must be integers." >&2

  exit 1

fi

sum=$(( a + b ))

echo "Sum: $sum"
```

8. What is the use of source command in shell scripting?

A.

The source command is used to execute a script within the current shell environment instead of starting a new shell.

- To run another script inside the same shell (so that variables, functions, or environment changes remain available after execution).
- Commonly used to load configuration files, set environment variables, or import functions into the current shell session.

9. How can you debug a shell script? Give two methods.

A.

Trace mode: show commands as they execute.

```
bash -x script.sh          # ad-hoc
# or inside script:
set -x   # enable
# ... code ...
set +x   # disable
# For better trace lines:
export PS4='+ ${BASH_SOURCE}:${LINENO}:${FUNCNAME[0]:-main}: '
```

Strict mode + fail fast: turn runtime mistakes into actionable errors.

```
set -euo pipefail
# -e: exit on error
# -u: fail on unset variables
# -o pipefail: pipeline fails if any command fails
```

Extras: print checkpoints, validate inputs with regex, and use shellcheck offline to catch common bugs.

10. Write a bash script to create and delete a file.

A.

```
#!/bin/bash
echo -n "Enter the file name: "
read filename
touch "$filename"
echo "File '$filename' created successfully."
 ls -l "$filename"
 echo -n "Do you want to delete this file? (y/n): "
read choice
if [ "$choice" = "y" ] || [ "$choice" = "Y" ]; then
   rm "$filename"
   echo "File '$filename' deleted successfully."
else
   echo "File '$filename' was not deleted."
Fi
```

Run:

```
chmod +x file_ops.sh
./file_ops.sh create ./tmp/demo.txt
./file_ops.sh delete ./tmp/demo.txt
```