

21. Write a shell script using if...else to check if a number is even or odd. ☺

This script asks the user to enter a number and checks if it's divisible by 2. If yes, it prints "Number is even"; otherwise, "Number is odd".

```
#!/bin/bash
echo "Enter a number:"
read num
if [ $((num % 2)) -eq 0 ]
then
echo "Number is even"
else
echo "Number is odd"
fi
```

22. Explain the difference between if and case statements in bash. ☺

if is mainly used to test conditions that result in true or false, handling simple yes/no checks. case is better when dealing with multiple possible values for a variable, like checking many options or menu choices, making the code neater than many if-elif tests.

23. Write a script to find the largest of three numbers entered by the user. ☺

This script takes three numbers as input and compares them step-by-step to find and print the largest:

```
#!/bin/bash
echo "Enter three numbers:"
read a b c
if [ $a -ge $b ] && [ $a -ge $c ]
then
echo "$a is largest"
elif [ $b -ge $a ] && [ $b -ge $c ]
then
echo "$b is largest"
else
echo "$c is largest"
fi
```

24. How do you use a for loop to traverse an array in bash? Give an example.

The array is defined as arr=(123, "Abs", -2.3, 'A', 23.56, 0). ☺

You can loop over every element in the array with this script:

```
#!/bin/bash
arr=(123 "Abs" -2.3 'A' 23.56 0)
```

```
for item in "${arr[@]}"
do
echo $item
done
```

25. Write a shell script to loop through all files in the current directory and display their names. ☺

This simple script prints the name of each file or folder in your current directory

```
#!/bin/bash
for file in *
do
echo $file
done
```

26. What is the difference between while and until loops in bash? ☺

A while loop continues running as long as its condition remains true. An until loop keeps going as long as its condition remains false and stops once the condition becomes true.

27. Write a countdown timer script using a while loop. ☺

This countdown script counts from 10 down to 1, waits one second at a time, then prints "Time's up!":

```
#!/bin/bash
count=10
while [ $count -gt 0 ]
do
echo "Countdown: $count"
sleep 1
((count--))
done
echo "Time's up!"
```

28. How do you use break and continue statements in loops? Give examples.

- break stops the entire loop immediately.
 - continue skips the current loop iteration and moves to the next one. ☺
- Example:

```
for i in {1..5}
do
```

```
if [ $i -eq 3 ]; then
    continue # Skip when i equals 3
fi
if [ $i -eq 4 ]; then
    break   # Stop the loop when i equals 4
fi
echo $i
done
```

29. Write a script to check if a file exists or not using the if and else loop.☒

This script asks for a filename and tells you whether the file exists:

```
#!/bin/bash
echo "Enter filename:"
read file
if [ -e $file ]
then
echo "File exists."
else
echo "File does not exist."
fi
```

30. Write a script to calculate factorial of a number using for loop.☒

This script calculates and prints the factorial of a given number:

```
#!/bin/bash
echo "Enter a number:"
read num
fact=1
for (( i=1; i<=num; i++ ))
do
fact=$((fact * i))
done
echo "Factorial of $num is $fact"
```