1. Write a shell script using if…else to check if a number is even or odd. (CO4)
Ans.

```
#!/bin/bash
echo "Enter a number:"

read num
if [ $((num % 2)) -eg 0]
then
  echo "The number is even."
else
  echo "The number is odd."
fi
```

2. Explain the difference between if and case statements in bash. (CO4)
Ans.

if	**case**
It is used for test conditions (true/false)	It matches a variable against different patterns
It can be used for logical comparisons	It is used for checking one variable against many possible values
It can handle complex conditions with elif	It can handle multiple conditions in a cleaner way
It is more flexible and has multiple use cases	It is limited to pattern matching
Example- checking if number is even/odd	Example- identify a fruit type from a list

3. Write a script to find the largest of three numbers entered by the user. (CO4)
Ans.

```bash
#!/bin/bash

echo "Enter first number:"
read num1
echo "Enter second number:"
read num2
echo "Enter third number:"
read num3

if [$num1 -ge $num2 ] && [$num1 -ge $num3 ]; then
  echo "The largest number is: $num1"


elif [$num2 -ge $num1 ] && [ $num2 -ge $num3 ]; then
  echo "The largest number is: $num2"
else
  echo "The largest number is: $num3"
fi
```

4. 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). (CO4)

Ans.

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

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

5. Write a shell script to loop through all files in the current directory and display their names. (CO4)

Ans.

```bash
#!/bin/bash
echo "Files in the current directory:"

for file in *
do
  if [ -f "$file" ]; then
    echo "$file"
  fi
done
```

6. What is the difference between while and until loops in bash? (CO4)

Ans.

while	until
It repeats a block of code while a condition is true	It repeats a block of code until a condition becomes true
The loop continues as long as the condition is true	The loop continues as long as the condition is false
The condition is checked before each iteration	The condition is checked before each iteration

7. Write a countdown timer script using a while loop. (CO4)

Ans.

```bash
#!/bin/bash
echo "Enter countdown time in seconds:"
read time

while [ $time -gt 0 ]
do
  echo "Time left: $time seconds"
  sleep 1
  ((time--))
done
echo "Time's up!"
```

8. How do you use break and continue statements in loops? Give examples. (CO4)

Ans.

Break

```
#!/bin/bash
for i in {1..5}

do
  if [ $i -eq 3 ]; then
    break
  fi
  echo "Number: $i"
done
```

```
priyanshu@priyanshu:~$ chmod +x break_continue.sh
priyanshu@priyanshu:~$ ./break_continue.sh
Number: 1
Number: 2
priyanshu@priyanshu:~$
```

Continue

```
#!/bin/bash
for i in {1..5}

do
  if [ $i -eq 3 ]; then
    continue
  fi
  echo "Number: $i"
done
```

```
priyanshu@priyanshu:~$ gedit break_continue.sh
priyanshu@priyanshu:~$ chmod +x break_continue.sh
priyanshu@priyanshu:~$ ./break_continue.sh
Number: 1
Number: 2
Number: 4
Number: 5
priyanshu@priyanshu:~$
```

9. Write a script to check if a file exists or not using the if and else loop. (CO4)
Ans.

```
#!/bin/bash
echo "Enter the file name:"

read filename
if [ -f "$filename" ]; then
  echo "File '$filename' exists."
else
  echo "File '$filename' does not exist."
fi

```

10. Write a script to calculate factorial of a number using for loop. (CO4)
Ans.

```
#!/bin/bash
echo "Enter a number: "

read num
factorial=1

for (( i=1; i<=num; i++))
do
  factorial=$((factorial * i))
done
echo "Factorial of $num is $factorial"
```