Linux Shell Scripting Lab – 20 Questions (Single-Page Markdown Format)

Q1. Complete: Even or Odd

```
[illegible]
[illegible] number"
el[illegible]
    echo "Odd number"
fi
```

Q2. Complete: Print 1 to 10

```
for i in {1...10}
do
    echo $i
done
```

Q3. Complete: File existence check

```
echo "Enter filename:"
read f
if [ ___ $f ]; then
    echo "File exists"
else
    echo "File not found"
fi
```

Q4. Complete: Compare two numbers

```
echo "Enter two numbers:"
read a b
if [ $a -gt $b ]; then
    echo "$a is greater"
else
    echo "$b is greater"
fi
```

Q5. Fill: Print current directory

```
echo "You are in directory:"
___
```

Q6. Debug: Numeric comparison error

```
a=10
b=5
if [ $a > $b ]; then
    echo "a is greater"
fi
```

Q7. Debug: read command error

```
echo "Enter your name:"
readname
echo "Hello $name"
```

Q8. Debug: while loop increment

```
i=1
while [ $i -le 5 ]
do
    echo $i
    i=$i+1
done
```

Q9. Debug: arithmetic expression

```
x=5
y=2
z= x + y
echo $z
```

Q10. Debug: case statement syntax

```
echo "Enter choice:"
read ch
case $ch in
1) echo "One"
2) echo "Two"
esac
```

Q11. Complete: Print numbers 1 to 20 using while loop

```
i=1
while [ $i -le 20 ]
do
    echo $i
    i=$((i+1))
done
```

Q12. Complete: Multiplication table (1–10) for a number

```
echo "Enter number:"
read n
for i in {1..10}
do
    echo "$n x $i = $((n*i))"
done
```

Q13. Complete: Factorial using for loop

```
echo "Enter number:"
read n
fact=1
for (( i=1; i<=n; i++ ))
do
    fact=$((fact*i))
done
echo "Factorial = $fact"
```

Q14. Complete: Sum of first 10 natural numbers

---

1) "if [$((n % 2)) -eq 0]; then" — 3
if the user's number is divisible by 2, then only we can say it's even, else it's odd.

2) "for i in {1...10}" — 5
we need do specify with bracket expansion do print the given set of numbers.

3) "if [-e $f]; then" — 2
it specifies that the file exists. What

4) "read a" "read b" — 0 / 5
we need to have separate variable for the program to run

5) "pwd"
prints user's current directory

6) "if [$a -gt $b]; then" — 5
In ~~bash script~~ shell scripting, "-gt" is used instead of ">"

7) "read name" — 5
"read" & "name" needs space between them for no syntax error.

8) "i=5
while [$i -gt 1]" — 0
While using 'while' loop, this way is much preferable.

9) "z= $((x+y))" — 3
In shell scripting, this way is used in order to get no error as this way the compiler will recognise which variable is used.

10) ";;" is missing after line 5 — 2

11) "echo "$i"" — 0
this is the only right statement [illegible] we can't echo [illegible] work.

12) It needs to be specified that 'n' belongs to {1...10} set as well "for n in {1...10} do" before line 3. — 0

```
do
    sum=$((sum + i))
done
echo "Sum = $sum"
```

Q15. Debug: List all files using for loop

```
for f in .
do
    echo $f
done
```

Q16. Debug: Check leap year (basic)

```
echo "Enter year:"
read y
if [ $((y%4)) -ne 0 ]; then
    echo "Not Leap"
else
    echo "Leap"
fi
```

Q17. Complete: Print even numbers 1–50 using while loop

```
i=2
while [ $i -le 50 ]
do
    echo $i
    i=$((i+2))
done
```

Q18. Complete: Count .sh files in current directory

```
count=0
for f in *.sh
do
    if [ -f "$f" ]; then
        count=$((count+1))
    fi
done
echo "Total .sh files: $count"
```

Q19. Complete: Read 5 numbers and print their sum

```
sum=0
for i in {1..5}
do
    read num
    sum=$((sum + num))
done
echo "Total = $sum"
```

Q20. Debug: Accept numbers until 0 entered then print sum

```
sum=0
while true
do
    read n
    if [ $n -eq 0 ]; then
        break
```

13) "echo "Factorial = $fact" (line 8) needs to use before "done" (line 7) in this code.

14) "echo "Sum = $sum" (line 6) needs to use before "done" (line 5) in this code.

15) "${[@]}"

16)

17) "echo "$i"" this is the only right way to use this echo statement, else it won't work.

18) line 8 should use before 7.

19) instead of 'read num' 'i' need to use specified variable by user num. "echo "Enter [illegible]