Shell Scripting 3

Q1.

```
#!/bin/bash

echo "Enter a number:"
read n

for ((i=1;i<=n;i++))
do
    for ((j=1;j<=i;j++))
    do
        echo -n "$j "
    done
    echo ""
done
"q1.sh" 14L, 149C                                                    14,0-1        All
```

```
root@DESKTOP-M3TSUJI:/home# cd infobell
root@DESKTOP-M3TSUJI:/home/infobell# vi q1.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q1.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q1.sh
Enter a number:
5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q2.

```
#!/bin/bash

echo "Enter a number:"
read n

count=1

for ((i=1;i<=n;i++))
do
    for ((j=1;j<=i;j++))
    do
        echo -n "$count "
        count=$((count+1))
    done
    echo ""
done
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q2.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q2.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q2.sh
Enter a number:
10
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54 55
root@DESKTOP-M3TSUJI:/home/infobell# ./q2.sh
Enter a number:
5
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q3.

```
#!/bin/bash

echo "Enter the first number:"
read num1

echo "Enter the second number:"
read num2

sum=$(echo "$num1 + $num2" | bc -l)

echo "The sum of $num1 and $num2 is $sum"
:wq
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q3.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q3.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q3.sh
Enter the first number:
4.28
Enter the second number:
1.21
The sum of 4.28 and 1.21 is 5.49
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q4.

```
#!/bin/bash

if [ $# -ne 3 ]
then
    echo "Usage: $0 <num1> <operator> <num2>"
    exit 1
fi

num1=$1
operator=$2
num2=$3

case $operator in
    "+")
        result=$(echo "$num1 + $num2" | bc -l)
        ;;
    "-")
        result=$(echo "$num1 - $num2" | bc -l)
        ;;
    "*")
        result=$(echo "$num1 * $num2" | bc -l)
        ;;
    "/")
        result=$(echo "scale=2; $num1 / $num2" | bc -l)
        ;;
    *)
        echo "Invalid operator. Valid operators are +, -, *, /."
        exit 1
        ;;
esac

echo "$num1 $operator $num2 = $result"

~
~
~
~
~
~
:wq
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q4.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q4.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q4.sh
Usage: ./q4.sh <num1> <operator> <num2>
root@DESKTOP-M3TSUJI:/home/infobell# vi q4.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q4.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q4.sh 1.2 + 2.6
1.2 + 2.6 = 3.8
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q5.

```
#!/bin/bash

if [ $# -eq 0 ]
then
    echo "Usage: $0 <num1> <num2> ... <numN>"
    exit 1
fi

max=$1

for num in "$@"
do
    if [ $num -gt $max ]
    then
        max=$num
    fi
done

echo "The largest number is: $max"

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q5.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q5.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q5.sh
Usage: ./q5.sh <num1> <num2> ... <numN>
root@DESKTOP-M3TSUJI:/home/infobell# ./q5.sh 1 3 8 6 5 7 9 2
The largest number is: 9
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q6.

```
#!/bin/bash

if [ $# -ne 1 ]
then
    echo "Usage: $0 <number>"
    exit 1
fi

number=$1
reversed=""

while [ $number -gt 0 ]
do
    digit=$(( $number % 10 ))
    reversed="$reversed$digit"
    number=$(( $number / 10 ))
done

echo "$1 in reverse order is $reversed"

~
~
~
~
~
~
~
~
:wq
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q6.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q6.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q6.sh 639872
639872 in reverse order is 278936
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q7.

```
#!/bin/bash

if [ $# -ne 1 ]
then
    echo "Usage: $0 <filename>"
    exit 1
fi

if [ ! -f $1 ]
then
    echo "$1 does not exist or is not a file"
    exit 1
fi

sed -i '/^$/d' $1

echo "Empty lines removed from $1"

~
~
~
~
~
~
~
~
~
:wq
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q7.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q7.sh
root@DESKTOP-M3TSUJI:/home/infobell# ls
f1  f2.txt.gz  f3.txt  q1.sh  q2.sh  q3.sh  q4.sh  q5.sh  q6.sh  q7.sh  t1
root@DESKTOP-M3TSUJI:/home/infobell# f3.txt
f3.txt: command not found
root@DESKTOP-M3TSUJI:/home/infobell# unzip f2.txt

Command 'unzip' not found, but can be installed with:

apt install unzip

root@DESKTOP-M3TSUJI:/home/infobell# gzip -d f2.txt
root@DESKTOP-M3TSUJI:/home/infobell# ls
f1  f2.txt  f3.txt  q1.sh  q2.sh  q3.sh  q4.sh  q5.sh  q6.sh  q7.sh  t1
root@DESKTOP-M3TSUJI:/home/infobell# cat f2.txt
Hello Infobell
root@DESKTOP-M3TSUJI:/home/infobell# ./q7.sh f2.txt
Empty lines removed from f2.txt
```

Q9

```
#!/bin/bash

read -p "Enter the value of n: " n

a=0
b=1

# first Fibonacci number
echo $a

# next Fibonacci numbers less than or equal to n
while [ $b -le $n ]
do
    echo $b
    c=$((a+b))
    a=$b
    b=$c
done

~
~
~
~
~
~
~
~
~
:wq
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q9.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q9.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q9.sh
Enter the value of n: 9
0
1
1
2
3
5
8
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q10.

```
#!/bin/bash

# initialize an array of strings
str_arr=("Hello" "world" "how" "are" "you")

# iterate through the array and print the length of each string
for s in "${str_arr[@]}"
do
    echo ${#s}
done

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q10.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q10.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q10.sh
5
5
3
3
3
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q11.

```
#!/bin/bash

# set the board size
board_size=8

# loop through each row of the board
for (( row=1; row<=board_size; row++ ))
do
    # loop through each column of the board
    for (( col=1; col<=board_size; col++ ))
    do
        # determine the value of the current square
        if (( ($row + $col) % 2 == 0 ))
        then
            val=0
        else
            val=1
        fi

        # print the value of the current square
        echo -n "$val "
    done

    # move to the next row
    echo ""
done

~
:wq
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q11.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q11.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q11.sh
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q12.

```
#!/bin/bash

read -p "Enter a list of numbers separated by spaces: " numbers
read -p "Enter sort order (asc/desc): " sort_order

if [ "$sort_order" == "asc" ]
then
    sorted_numbers=$(echo "$numbers" | tr ' ' '\n' | sort -n)
elif [ "$sort_order" == "desc" ]
then
    sorted_numbers=$(echo "$numbers" | tr ' ' '\n' | sort -nr)
else
    echo "Invalid sort order"
    exit 1
fi

echo "Sorted numbers: $sorted_numbers"


~
~
~
~
~
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q12.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q12.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q12.sh
Enter a list of numbers separated by spaces: 96 12 85 36 25 1 9 14 14 11
Enter sort order (asc/desc): asc
Sorted numbers: 1
9
11
12
14
14
25
36
85
96
```

Q30.

```
#!/bin/bash

# awk to filter and count the number of users with user IDs between 500 and 10000
count=$(awk -F: '$3 >= 500 && $3 <= 10000 {print}' /etc/passwd | wc -l)

echo "Number of users with user IDs between 500 and 10000: $count"

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"q30.sh" 7L, 236C                                                    3,2           All
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q30.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q30.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q30.sh
Number of users with user IDs between 500 and 10000: 2
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q28.

```
#!/bin/bash

directory="/home/education"

# setting the permissions to be locked
permissions="go-rwx"

# setting the permissions recursively for all files and directories
chmod -R $permissions $directory

echo "Permissions locked for directory $directory"

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"q28.sh" 12L, 257C                                                   3,26          All
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q28.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q28.sh
Permissions locked for directory /home/education
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q17.

```
#!/bin/bash

# read input from the user
read -p "Enter the file path: " filepath
read -p "Enter the starting line number: " startline
read -p "Enter the number of lines to print: " numlines

# print the requested lines using sed
sed -n "$startline,$((startline+numlines-1))p" "$filepath"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"q17.sh" 10L, 289C                                                    10,0-1        All
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q17.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q17.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q17.sh
Enter the file path: /home/education
Enter the starting line number: 1
Enter the number of lines to print: 6
sed: read error on /home/education: Is a directory
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q18.

```
#!/bin/bash

# get the longest and shortest user-names using awk
longest=$(awk -F: 'length($1) > length(longest) {longest = $1} END {print longest}' /etc/passwd)
shortest=$(awk -F: 'length($1) < length(shortest) || !shortest {shortest = $1} END {print shortest}' /etc/passwd)

# display the results
echo "Longest username: $longest"
echo "Shortest username: $shortest"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"q18.sh" 11L, 371C                                                    11,0-1        All
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q18.h
root@DESKTOP-M3TSUJI:/home/infobell# vi q18.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q18.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q18.sh
Longest username: systemd-timesync
Shortest username: lp
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q19.

```
#!/bin/bash

dir="/home/education"

# used find command to search for .swp files in the specified directory
# and deleted them using rm command
find "$dir" -type f -name "*.swp" -exec rm -f {} \;

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    2,1           All
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q19.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q19.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q19.sh
root@DESKTOP-M3TSUJI:/home/infobell#
```

Q20.

```
#!/bin/bash

# generation of random 8-character password
password=$(head /dev/urandom | tr -dc A-Za-z0-9 | head -c 8 ; echo '')

# display the password
echo "Your new password is: $password"

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"q20.sh" 8L, 192C                                               3,15          All
```

```
root@DESKTOP-M3TSUJI:/home/infobell# vi q20.sh
root@DESKTOP-M3TSUJI:/home/infobell# chmod +x q20.sh
root@DESKTOP-M3TSUJI:/home/infobell# ./q20.sh
Your new password is: GY3zCyA2
root@DESKTOP-M3TSUJI:/home/infobell#
```

Shell Script 1

- Create the startup script for an application start and stop.

```
root@DESKTOP-M3TSUJI:/home# vi script1.sh
```

```
#!/bin/bash

APP_NAME="script1"
APP_PATH="C:\Users\Simran\Infobell"

start() {
    if pgrep -f "$APP_NAME" > /dev/null; then
        echo "$APP_NAME is already running"
    else
        cd "$APP_PATH"
        nohup ./run.sh > /dev/null 2>&1 &
        echo "Started $APP_NAME"
    fi
}

stop() {
    if pgrep -f "$APP_NAME" > /dev/null; then
        pkill -f "$APP_NAME"
        echo "Stopped $APP_NAME"
    else
        echo "$APP_NAME is not running"
    fi
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        sleep 1
        start
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
        ;;
esac

exit 0
~
~
~
```

```
root@DESKTOP-M3TSUJI:/home# chmod +x script1.sh
root@DESKTOP-M3TSUJI:/home# ./script1.sh start
script1 is already running
root@DESKTOP-M3TSUJI:/home# ./script1.sh stop
Terminated
root@DESKTOP-M3TSUJI:/home# ./script1.sh restart
Terminated
```

- Write a shell script that consists of a function that displays the number of files in the present working directory. Name this function “file_count” and call it in your script. If you use variable in your function, remember to make it a local variable.

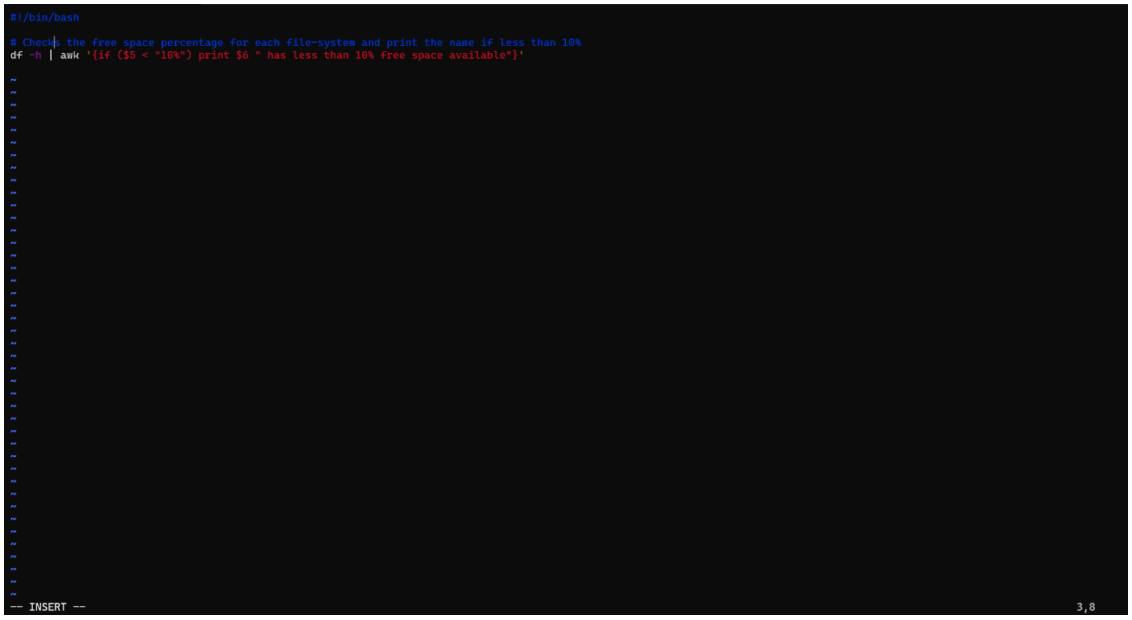
```
root@DESKTOP-M3TSUJI:/home# vi script2.sh
```

```
#!/bin/bash

file_count () {
    local count =$(ls -1 | wc -l)
    echo "$count number of files are present in the directory"
}

file_count

-- INSERT --                                          7,1           All
```

```
root@DESKTOP-M3TSUJI:/home# vi script2.sh
root@DESKTOP-M3TSUJI:/home# chmod +x script2.sh
root@DESKTOP-M3TSUJI:/home# ./script2.sh
./script2.sh: line 4: local: `=33': not a valid identifier
 number of files are present in the directory
```

- For each directory in the $PATH, display the number of executable in that directory

- Display the names of any file-system which have less than 10% free space available



- Write a script that takes any number of directories as command-line arguments and then lists the contents of each of these directories.