Practise Questions Shell Scripting

Ques: To read n and generate a below pattern

```
1
1 2
1 2 3
1 2 3 4
```

Input- bash 01_pattern.sh
Enter the value of n:6

Output- The expected pattern is

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

```
 GNU nano 6.2
#!/bin/bash
echo "Enter the value of n : "
read n
echo "The expected pattern is: "
for((i==1;i<=n;i++))
do
        for((j=1;j<=i;j++))
        do
                echo -n "$j"
        done
        echo""
done
```

```
srishtirupa@linux:~$ nano s1.sh
srishtirupa@linux:~$ chmod 777 s1.sh
srishtirupa@linux:~$ bash s1.sh
Enter the value of n :
5
The expected pattern is:

-n1

-n1
-n2

-n1
-n2
-n3

-n1
-n2
-n3
-n4

-n1
-n2
-n3
-n4
-n5

srishtirupa@linux:~$ nano s1.sh
srishtirupa@linux:~$ bash s1.sh
Enter the value of n :
5
The expected pattern is:

1
12
123
1234
12345
srishtirupa@linux:~$ nano s1.sh
srishtirupa@linux:~$
```

Ques : Description-To read n and generate a below pattern

1
2 3
4 5 6
7 8 9 10

Input- bash 02_pattern.sh
Enter the value of n:4
Output- The expected pattern is

1
2 3
4 5 6
7 8 9 10

```
  GNU nano 6.2
#!/bin/bash

echo "Enter the value of n : "

read n

echo "The expected pattern is: "

num=1
row=1
while[ $row -le $n ]
do

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

```
srishtirupa@linux:~$ nano s2.sh
srishtirupa@linux:~$ bash s2.sh
Enter the value of n:
4
The expected paΣern is:
1
2 3
4 5 6
7 8 9 10
srishtirupa@linux:~$
```

Ques : Description-Script for addition of two real numbers
Input- bash 03_real_add.sh
Enter the numbers to addition: 4.28 1.21
Output- The sum of 4.28 and 1.21 is 5.49

```
  GNU nano 6.2
#!/bin/bash
echo "Enter the numbers for  addition : "
read n1 n2
sum=$(echo "$n1 + $n2" |bc)
echo "sum is $sum"
```

```
srishtirupa@linux:~$ touch s3.sh
srishtirupa@linux:~$ chmod 777 s3.sh
srishtirupa@linux:~$ nano s3.sh
srishtirupa@linux:~$ bash s3.sh
Enter the numbers for  addition :
2 3
sum is 5
srishtirupa@linux:~$
```

Ques : Description- Script for arithmetic calculator using command line arguments

Input- bash 04_calculator.sh 1.2 + 2.6

Output- The sum of 1.2 and 2.6 is 3.8

```
srishtirupa@linux:~$ nano a1.sh
srishtirupa@linux:~$ ./a1.sh
./a1.sh: line 4: -h: command not found

srishtirupa@linux:~$ nano a1.sh
srishtirupa@linux:~$ ./a1.sh
./a1.sh: line 4: var: command not found

srishtirupa@linux:~$ nano a1.sh
srishtirupa@linux:~$ ./a1.sh
tmpfs
tmpfs
tmpfs
tmpfs

srishtirupa@linux:~$ ./a1.sh
```

```
#!/bin/bash
if[ $# -ne 3 ];
then
        echo "Usage:$0 n1 operator n2"
        exit 1
fi

n1=$1
operator=$2
n2=$3
case $operator in
        +)
                result=$(echo "$n1 + $n2" |bc)
                ;;
        -)
                result=$(echo "$n1 - $n2" |bc)
                ;;
        \*)
                result=$(echo "$n1 * $n2" |bc)
                ;;
        /)
                result=$(echo "$n1 / $n2" |bc)
                ;;
        *)
                echo "Invalid Operator"
        exit 1
esac
echo "the $operator of $n1 and $n2 is $result"
```

Ques : Description- Script to compare larger integer values from a 'n' number of arguments using command line arguments
Input- bash 05_largest.sh 1 3 8 6 5 7 9 2
Output- The largest value is 9

```
#!/bin/bash
# Check if arguments are provided
if [ $# -eq 0 ]; then
echo "Usage: $0 arg1 arg2 ... argn"
exit 1
fi
# Set the first argument as the current largest value
largest=$1
# Loop through the remaining arguments and compare with the current largest value
for arg in "$@"; do
if [ "$arg" -gt "$largest" ]; then
largest=$arg
fi
done
# Print the largest value
echo "The largest value is $largest"
```

```
srishtirupa@linux:~$ touch s5.sh
srishtirupa@linux:~$ chmod 777 s5.sh
srishtirupa@linux:~$ nano s5.sh
srishtirupa@linux:~$ bash s5.sh 2 3 1 4
s5.sh: line 1: !/bin/bash: No such file or directory
The largest value is 4
srishtirupa@linux:~$ nano s5.sh
srishtirupa@linux:~$ bash s5.sh 2 3 1 4
The largest value is 4
srishtirupa@linux:~$ bash s5.sh 2 3 1 4 46 2 7
The largest value is 46
srishtirupa@linux:~$
```

Ques : Description- Script to print a given number in reverse order.
Input- bash 06_reverse.sh 639872
Output- The reversed number of entered number is 278936

```
#!/bin/bash
# Check if an argument is provided
if [ $# -eq 0 ]; then
echo "Usage: $0 number"
exit 1
fi
# Store the argument in a variable
number=$1
# Reverse the number using a loop
reverse=""
while [ "$number" -gt 0 ]; do
remainder=$(( $number % 10 ))
reverse="$reverse$remainder"
number=$(( $number / 10 ))
done
# Print the reversed number
echo "The reversed number of entered number is $reverse"
```

```
srishtirupa@linux:~$ touch s6.sh
srishtirupa@linux:~$ chmod 777 s6.sh
srishtirupa@linux:~$ nano s6.sh
srishtirupa@linux:~$ bash s6.sh 123
The reversed number of entered number is 321
srishtirupa@linux:~$
```

Ques : Description- Script to delete empty lines from a file
Input- bash 07_delete_empty_lines.sh file.txt
Output- All empty lines of the file file.txt will be deleted
Before script running, content of the file file.txt :
Hello, I am Siddaling.

I am from Belgaum.

I studied B-Tech in ECE.
After script running, content of the file file.txt :
Hello, I am Siddaling.
I am from Belgaum.
I studied B-Tech in ECE.

```
#!/bin/bash

if [ $# -eq 0 ]; then
echo "Usage: $0 filename"
exit 1
fi

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

sed -i '/^\s*$/d' "$1"

echo "All empty lines of the file $1 will be deleted"

echo "Before script running, content of the file $1 :"
cat "$1"
echo "Afer script running, content of the file $1 :"
cat "$1"
```

```
srishtirupa@linux:~$ touch s7.sh
srishtirupa@linux:~$ chmod 777 s7.sh
srishtirupa@linux:~$ nano s7.sh
srishtirupa@linux:~$ ls
a1.sh          Demo.txt      msg.txt    s3.sh            snap
BigData        demo.txt.gz   Music      s4.sh            T1.txt
class.txt      Desktop       new.txt    s5.sh            Templates
copied13.txt   Documents     Pictures   s6.sh            test12.txt
demo1          Downloads     Public     s7.sh            test13.txt
demo12.txt     igTest        s1.sh      secondDir,hello  test3
demo2.txt.gz   marks.txt     s2.sh      shreya1          Videos
srishtirupa@linux:~$ bash s7.sh class.txt
All empty lines of the file class.txt will be deleted
Before script running, content of the file class.txt :
Sam cs
Daniel cs
John IT
Arya IT
Mike ECE
Afer script running, content of the file class.txt :
Sam cs
Daniel cs
John IT
Arya IT
Mike ECE
```

Ques :Description- Script to perform arithmetic operation on digits of a given number depending upon the operator.

Input- bash 08_operator_dependent.sh 12354+
Output- The sum is 15

```
srishtirupa@linux:~$ touch s8.sh
srishtirupa@linux:~$ chmod 777 s8.sh
srishtirupa@linux:~$ nano s8.sh
srishtirupa@linux:~$ bash s8.sh 12345+
The + of the digits in 12345 is 15
srishtirupa@linux:~$ bash s8.sh 12345\*
The * of the digits in 12345 is 0
srishtirupa@linux:~$ bash s8.sh 12345*
The * of the digits in 12345 is 0
srishtirupa@linux:~$ bash s8.sh 12345-
The - of the digits in 12345 is -15
srishtirupa@linux:~$ bash s8.sh 12345/
The / of the digits in 12345 is 0
srishtirupa@linux:~$
```

```bash
#!/bin/bash

if [ $# -eq 0 ]; then
echo "Usage: $0 string"
exit 1
fi

number=${1%[-+*/]}
operator=${1:${#number}:1}

if [[ ! $operator =~ [-+*/] ]]; then
echo "Error: Invalid operator."
exit 1
fi
result=0
for ((i=0; i<${#number}; i++)); do
digit=${number:$i:1}
case $operator in
+) result=$((result + digit)) ;;
-) result=$((result - digit)) ;;
\*) result=$((result * digit)) ;;
/) result=$((result / digit)) ;;
esac
done

echo "The $operator of the digits in $number is $result"
```

Ques : Description- script to read 'n' and generate Fibonacci numbers <= n
Input- bash 09_fibonacci.sh
Enter limit for fibonacci series: 13
Output- The expected fibonacci series is:
0, 1, 1, 2, 3, 5, 8, 13,

```bash
#!/bin/bash

read -p "Enter limit for Fibonacci series: " n
a=0
b=1

echo -n "$a"

while [ $b -le $n ]; do
echo -n ", $b"
c=$((a + b))
a=$b
b=$c
done
echo ""
```

```
srishtirupa@linux:~$ touch s9.sh
srishtirupa@linux:~$ chmod 777 s9.sh
srishtirupa@linux:~$ nano s9.sh
srishtirupa@linux:~$ nano s9.sh
srishtirupa@linux:~$ bash s9.sh 13
Enter limit for Fibonacci series: 13
0, 1, 1, 2, 3, 5, 8, 13
srishtirupa@linux:~$ bash s9.sh
Enter limit for Fibonacci series: 78
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55
srishtirupa@linux:~$
```

Ques : Description- Script to print the length of each and every string using arrays

Input- bash 10_string_length.sh hello, I am Siddaling

Output- The lengths of each string are as below:
length of the string(hello,) -6
length of the string(I) -1
length of the string(am) -2
length of the string(Siddaling) -9

```
#!/bin/bash

read -p "Enter a string: " input_string

string_array=($input_string)

echo "The lengths of each string are as below:"
for word in "${string_array[@]}"; do
echo "Length of the string($word) -${#word}"
done
```

```
srishtirupa@linux:~$ touch s10.sh
srishtirupa@linux:~$ chmod 777 s10.sh
srishtirupa@linux:~$ nano s10.sh
srishtirupa@linux:~$ bash s10.sh
Enter a string: 1233576588
The lengths of each string are as below:
Length of the string(1233576588) -10
srishtirupa@linux:~$ bash s10.sh
Enter a string: 124235 dfshg 56g hfdh
The lengths of each string are as below:
Length of the string(124235) -6
Length of the string(dfshg) -5
Length of the string(56g) -3
Length of the string(hfdh) -4
srishtirupa@linux:~$
```

Ques : Description- script to print chess board , black as 1 , white as 0
Input- bash 11_chess_board.sh
Output- [Chess board]

```
#!/bin/bash

rows=8
cols=8

for (( row=0; row<$rows; row++ )); do
for (( col=0; col<$cols; col++ )); do

if (( ($row+$col) % 2 == 0 )); then
echo -n "1 "
else
echo -n "0 "
fi
done
echo ""
done
```

```
srishtirupa@linux:~$ touch s11.sh
srishtirupa@linux:~$ chmod 777 s11.sh
srishtirupa@linux:~$ nano s11.sh
srishtirupa@linux:~$ bash s11.sh
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
0 1 0 1 0 1 0 1
srishtirupa@linux:~$
```

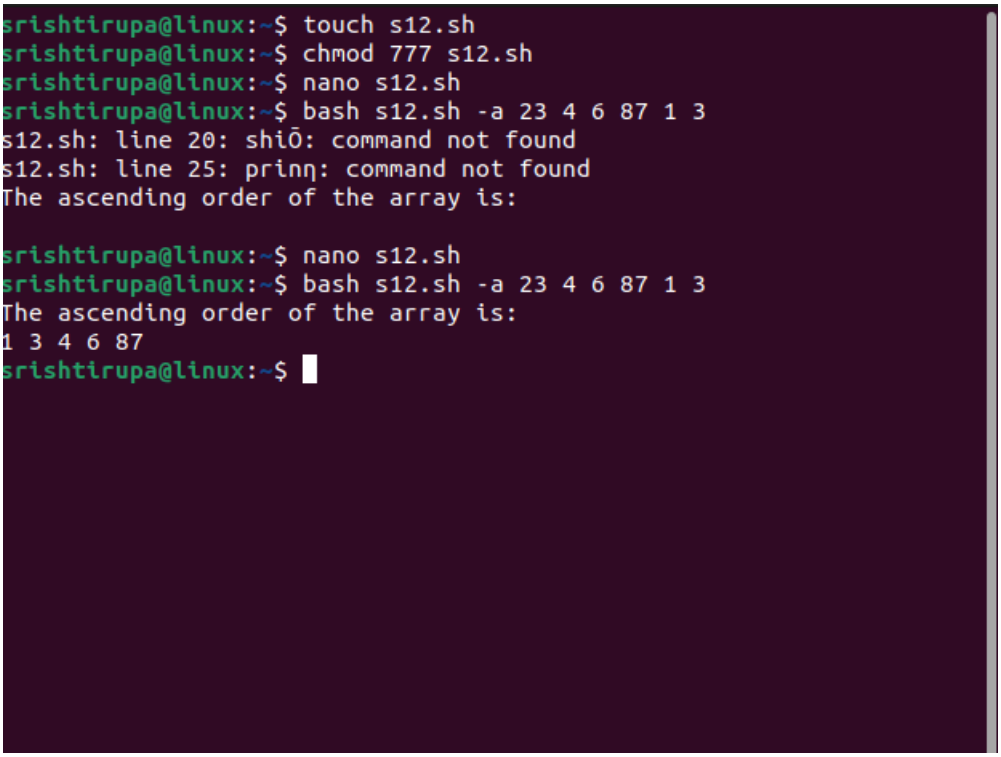
Ques : Description- Script to sort a given number in ascending or descending order.
Input- bash 12_sorting.sh -a 96 12 85 36 25 1 9 14 14 11
Output- The ascending order of the array is:
1 9 11 12 14 14 25 36 85 96

```
  GNU nano 6.2                    s12.sh
#!/bin/bash
while getopts "a:d:" opt; do
case ${opt} in
a )
order="ascending"
;;
d )
order="descending"
;;
\? )
echo "Invalid option: -$OPTARG" 1>&2
exit 1
;;
: )
echo "Option -$OPTARG requires an argument." 1>&2
exit 1
;;
esac
done
shift $((OPTIND -1))

args=("$@")

if [[ "$order" == "ascending" ]]; then
sorted_args=($(printf '%s\n' "${args[@]}" | sort -n))
elif [[ "$order" == "descending" ]]; then
sorted_args=($(printf '%s\n' "${args[@]}" | sort -nr))
fi

echo "The $order order of the array is: "
echo "${sorted_args[@]}"
```



Ques : Description- Script to print the following:

- Currently logged users
- Your shell directory
- Home directory
- OS name & version
- Current working directory
- Number of users logged in
- Show all available shells in your system
- Hard disk information
- CPU information.
- Memory information.
- File system information.
- Currently running process

Input- bash 13_system_info.sh

1. Currently logged users
2. Your shell directory
3. Home directory
4. OS name & version
5. Current working directory
6. Number of users logged in
7. Show all available shells in your system
8. Hard disk information
9. CPU information.

10.Memory information.
11.File system information.
12.Currently running process.
Enter the option: 3

```
#!/bin/bash

while true; do
    echo "1. Currently logged users"
    echo "2. Your shell directory"
    echo "3. Home directory"
    echo "4. OS name & version"
    echo "5. Current working directory"
    echo "6. Number of users logged in"
    echo "7. Show all available shells in your system"
    echo "8. Hard disk information"
    echo "9. CPU information"
    echo "10. Memory information"
    echo "11. File system information"
    echo "12. Currently running process"
    echo "Enter the option:"
    read option

    case $option in
        1)
            who
            ;;
        2)
            echo "Your shell directory is $SHELL"
            ;;
        3)
            echo "Your home directory is $HOME"
            ;;
        4)
            echo "OS name & version:"
            uname -a
            ;;
        5)
            echo "Current working directory:"
            pwd
            ;;
        6)
            echo "Number of users logged in:"
            who | wc -l
            ;;
```

```
        ;;
    6)
        echo "Number of users logged in:"
        who | wc -l
        ;;
    7)
        echo "Available shells:"
        cat /etc/shells
        ;;
    8)
        echo "Hard disk information:"
        df -h
        ;;
    9)
        echo "CPU information:"
        lscpu
        ;;
    10)
        echo "Memory information:"
        free -h
        ;;
    11)
        echo "File system information:"
        df -T
        ;;
    12)
        echo "Currently running process:"
        ps
        ;;
    *)
        echo "Invalid option"
        ;;
esac
done
```

```
srishtirupa@linux:~$ touch s131.sh
srishtirupa@linux:~$ chmod 777 s131.sh
srishtirupa@linux:~$ nano s131.sh
srishtirupa@linux:~$ bash s131.sh
1. Currently logged users
2. Your shell directory
3. Home directory
4. OS name & version
5. Current working directory
6. Number of users logged in
7. Show all available shells in your system
8. Hard disk information
9. CPU information
10. Memory information
11. File system information
12. Currently running process
Enter the option:
3
Your home directory is /home/srishtirupa
1. Currently logged users
2. Your shell directory
3. Home directory
4. OS name & version
5. Current working directory
6. Number of users logged in
7. Show all available shells in your system
8. Hard disk information
9. CPU information
10. Memory information
11. File system information
12. Currently running process
Enter the option:
4
OS name & version:
Linux linux 5.19.0-38-generic #39~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Mar 17 21:16:15 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
```

Ques : Description- Script to rename a file/directory replaced by lower/upper case letters.

Input- bash 14_file_upper_lower.sh

Output-Before running the script

ls

File.txt MyScript.SH MyFile007.txt dir/ Assign1/ newfolder/

After running the script

$ ls

file.txt myfile007.txt myscript.sh DIR/ ASSIGN1/ NEWFOLDER/

```
#!/bin/bash

echo "Enter the directory path:"
read directory

if [[ "$file" == "${file^^}" ]];then

# to lowercase
for file in "$directory"/*; do
    if [[ -f "$file" ]]; then
        mv "$file" "${file,,}"
    elif [[ -d "$file" ]]; then
        mv "$file" "${file,,}"
    fi
done
elif [[ "$file" == "${file,,}" ]];then

# to uppercase
for file in "$directory"/*; do
    if [[ -f "$file" ]]; then
        mv "$file" "${file^^}"
    elif [[ -d "$file" ]]; then
        mv "$file" "${file^^}"
    fi
done
fi
echo "After running the script:"
ls "$directory"
```

```
srishtirupa@linux:~$ touch s141.sh
srishtirupa@linux:~$ chmod 777 s141.sh
srishtirupa@linux:~$ nano s141.sh
srishtirupa@linux:~$ bash s141.sh
Enter the directory path:
,
After running the script:
ls: cannot access ',': No such file or directory
srishtirupa@linux:~$ bash s141.sh
Enter the directory path:
.
mv: './s131.sh' and './s131.sh' are the same file
mv: './s141.sh' and './s141.sh' are the same file
mv: './s14.sh' and './s14.sh' are the same file
mv: './s15.sh' and './s15.sh' are the same file
mv: './shell scripting.txt' and './shell scripting.txt' are the same file
mv: cannot move './snap' to a subdirectory of itself, './snap/snap'
After running the script:
 A1.SH          DEMO12.TXT     DOCUMENTS   MUSIC      S11.SH    S14.SH   S4.SH   S9.SH                 T1.TXT       VIDEOS
 BIGDATA        DEMO2.TXT.GZ   DOWNLOADS   NEW.TXT    S12.SH    S15.SH   S5.SH   SECONDDIR,HELLO       TEMPLATES
 CLASS.TXT      DEMO.TXT       IGTEST      PICTURES   S131.SH   S1.SH    S6.SH  'SHELL SCRIPTING.TXT'  TEST12.TXT
 COPIED13.TXT   DEMO.TXT.GZ    MARKS.TXT   PUBLIC     S13.SH    S2.SH    S7.SH   SHREYA1               TEST13.TXT
 DEMO1          DESKTOP        MSG.TXT     S10.SH     S141.SH   S3.SH    S8.SH   SNAP                  TEST3
```

Ques : Description- Script to rename current working directory with given name.
Input- bash 15_rename_cur_dir.sh siddaling
Output- Before running the script:
Name of current directory- siddaling1
After running the script:
Name of current directory- siddaling

```
#!/bin/bash

echo "Enter the new name for the current working directory:"
read newname


curdir=$(basename "$PWD")


if [ "$curdir" != "$newname" ]; then
    mv "$PWD" "${PWD%/*}/$newname"
fi

echo "Name of current directory- $(basename "$PWD")"
```

```
srishtirupa@linux:~$ ls
 a1.sh          demo12.txt     documents   music      s11.sh    s14.sh   s4.sh   s9.sh                 t1.txt      videos
 bigdata        demo2.txt.gz   downloads   new.txt    s12.sh    s15.sh   s5.sh   seconddir,hello       templates
 class.txt      demo.txt       igtest      pictures   s131.sh   s1.sh    s6.sh  'shell scripting.txt'  test12.txt
 copied13.txt   demo.txt.gz    marks.txt   public     s13.sh    s2.sh    s7.sh   shreya1               test13.txt
 demo1          desktop        msg.txt     s10.sh     s141.sh   s3.sh    s8.sh   snap                  test3
srishtirupa@linux:~$ s151.sh
^[[As151.sh: command not found
srishtirupa@linux:~$ touch s151.sh
srishtirupa@linux:~$ chmod 777 s151.sh
srishtirupa@linux:~$ nano s151.sh
srishtirupa@linux:~$ bash s151.sh
Enter the new name for the current working directory:
shrey
mv: cannot move '/home/srishtirupa' to '/home/shrey': Permission denied
Name of current directory- srishtirupa
srishtirupa@linux:~$
```

Ques : Description- Script to rename all .jpg files by replacing prefix which is given by user
Input- bash 16_rename_album.sh myday
Output- Before running the script
ls

16_rename_album.sh DSN001.jpg DSN002.jpg DSN003.jpg DSN004.jpg DSN005.jpg DSN006.jpg DSN007.jpg

After running the script

$ ls

16_rename_album.sh myday_001.jpg myday_002.jpg myday_003.jpg myday_004.jpg myday_005.jpg myday_006.jpg myday_007.jpg

```
#!/bin/bash

echo "Enter the new prefix for .jpg files:"
read newprefix

for file in *.jpg; do
    newname="${newprefix}_${file}"
    mv "$file" "$newname"
done

echo "After running the script:"
ls
```

```
srishtirupa@linux:~$ touch s161.sh
srishtirupa@linux:~$ chmod 777 s161.sh
srishtirupa@linux:~$ nano s161.sh
srishtirupa@linux:~$ touch 1.jpg s.jpg
srishtirupa@linux:~$ ls
 1.jpg          demo12.txt     downloads   pictures    s13.sh    s1.sh    s7.sh                 s.jpg        test3
 a1.sh          demo2.txt.gz   igtest      public      s141.sh   s2.sh    s8.sh                 snap         videos
 bigdata        demo.txt       marks.txt   s10.sh      s14.sh    s3.sh    s9.sh                 t1.txt
 class.txt      demo.txt.gz    msg.txt     s11.sh      s151.sh   s4.sh    seconddir,hello       templates
 copied13.txt   desktop        music       s12.sh      s15.sh    s5.sh  'shell scripting.txt'   test12.txt
 demo1          documents      new.txt     s131.sh     s161.sh   s6.sh    shreya1               test13.txt
srishtirupa@linux:~$ bash s161.sh
Enter the new prefix for .jpg files:
new
After running the script:
 a1.sh          demo12.txt     documents   music       public    s13.sh    s161.sh   s5.sh   seconddir,hello       templates
 bigdata        demo2.txt.gz   downloads   new_1.jpg   s10.sh    s141.sh   s1.sh     s6.sh  'shell scripting.txt'  test12.txt
 class.txt      demo.txt       igtest      new_s.jpg   s11.sh    s14.sh    s2.sh     s7.sh   shreya1               test13.txt
 copied13.txt   demo.txt.gz    marks.txt   new.txt     s12.sh    s151.sh   s3.sh     s8.sh   snap                  test3
 demo1          desktop        msg.txt     pictures    s131.sh   s15.sh    s4.sh     s9.sh   t1.txt                videos
srishtirupa@linux:~$
```

Ques : Description- Script to print contents of file from given line number to next given number of lines.

Input- bash 17_print_lines.sh 5 4

Output- line 5

line 6

line 7

line 8

```
#!/bin/bash

line_number=3
number_of_lines=5

# Print the selected lines
sed -n "${line_number},${line_number + $number_of_lines - 1}p" $0
```

Ques : Description- Script to display the longest and shortest user-names on the system.
Input- bash 18_largest_username.sh
Output- The user with longest name is gnome-initial-setup
The user with shortest name is lp

```
#!/bin/bash


longest=$(awk -F: '{print length($1) " " $1}' /etc/passwd | sort -n | tail -n 1 | awk '{print $2}')
shortest=$(awk -F: '{print length($1) " " $1}' /etc/passwd | sort -n | head -n 1 | awk '{print $2}')


echo "The user with longest name is $longest"
echo "The user with shortest name is $shortest"
```

```
srishtirupa@linux:~$ ls
 a1.sh          demo2.txt.gz   igtest      new.txt    s131.sh   s161.sh   s5.sh            'shell scripting.txt'   test13.txt
 bigdata        demo.txt       marks.txt   pictures   s13.sh    s171.sh   s6.sh             shreya1                test3
 class.txt      demo.txt.gz    msg.txt     public     s141.sh   s1.sh     s7.sh             snap                   videos
 copied13.txt   desktop        music       s10.sh     s14.sh    s2.sh     s8.sh             t1.txt
 demo1          documents      new_1.jpg   s11.sh     s151.sh   s3.sh     s9.sh             templates
 demo12.txt     downloads      new_s.jpg   s12.sh     s15.sh    s4.sh     seconddir,hello   test12.txt
srishtirupa@linux:~$ touch s18
srishtirupa@linux:~$ touch s18.sh
srishtirupa@linux:~$ chmod 777 s18.sh
srishtirupa@linux:~$ nano s18.sh
srishtirupa@linux:~$ bash s18.sh
The user with longest name is gnome-initial-setup
The user with shortest name is lp
srishtirupa@linux:~$
```

Ques : Description- Script to delete all the .swp files found in your system or directory.
Input- bash 19_delete_display_swp.sh

Output- swp files found:

./b.swp
./siddaling/b.swp
./siddaling/c.swp
./siddaling/test/b.swp
./siddaling/test/c.swp
./siddaling/test/d.swp
./siddaling/test/a.swp
./siddaling/test/e.swp
./siddaling/d.swp
./siddaling/a.swp
./a.swp

```
echo "swp files found:"
find / -type f -name "*.swp" 2>/dev/null | tee /dev/tty


read -p "Do you want to delete these files? [y/n]: " answer
if [[ $answer == "y" ]]; then

    find / -type f -name "*.swp" -delete 2>/dev/null
    echo "swp files deleted."
fi
```

```
srishtirupa@linux:~$ touch s19.sh
srishtirupa@linux:~$ chmod 777 s19.sh
srishtirupa@linux:~$ nano s19.sh
srishtirupa@linux:~$ bash s19.sh
swp files found:
/home/srishtirupa/.demo3.txt.swp
/home/srishtirupa/.demo3.txt.swp
```

Ques : Description- script for generating random 8-character passwords including alpha numeric characters.

Input- bash 20_random_password.sh

Output- The generated 8 random passwords are:

J,kwAc{O
Pft4Get*
\2('pKlr
!,[(VwY%
^X|Z+}u8
y]\(2P|R

B#6'V=#D
8\Lrz{bc

```
#!/bin/bash

function generate_password() {
    local length=$1

    local chars='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'

    for i in $(seq 1 $length); do
        echo -n "${chars:RANDOM%${#chars}:1}"
    done
    echo
}


echo "The generated 8 random passwords are:"
for i in $(seq 1 8); do
    generate_password 8
done
```

```
srishtirupa@linux:~$ touch s20.sh
srishtirupa@linux:~$ chmod 777 s20.sh
srishtirupa@linux:~$ nano s20.sh
srishtirupa@linux:~$ bash s20.sh
The generated 8 random passwords are:
1a5Y05kz
oZYuHlOg
zb1ZGlUv
BKl4YlIV
T0uaRBoj
wu1nj1za
QclfVFLZ
8GIiPwzt
srishtirupa@linux:~$
```

Ques : Description- Script called say_hello, which will print greetings based on time and to provide date information .
Input- in bashrc file- /home/siddaling/21_say_hello.sh
Output- Good evening siddaling,have nice day!
This is Saturday 09 in January of 2021 (07:58:53 PM)

```bash
#!/bin/bash

hour=$(date +"%H")

user=$(whoami)

if [ $hour -ge 6 -a $hour -lt 12 ]; then
    echo "Good morning $user, have a nice day!"

elif [ $hour -ge 12 -a $hour -lt 18 ]; then
    echo "Good afternoon $user, have a nice day!"

else
    echo "Good evening $user, have a nice day!"
fi

date +"This is %A %d in %B of %Y (%r)"
```

```
srishtirupa@linux:~$ touch s21.sh
srishtirupa@linux:~$ chmod s21.sh
chmod: missing operand after ‘s21.sh’
Try 'chmod --help' for more information.
srishtirupa@linux:~$ chmod 777 s21.sh
srishtirupa@linux:~$ nano s22.sh
srishtirupa@linux:~$ nano s21.sh
srishtirupa@linux:~$ bash s21.sh
Good afternoon srishtirupa, have a nice day!
This is Thursday 20 in April of 2023 (04:26:03 PM IST)
srishtirupa@linux:~$
```

Ques : Description- Script to use pipes or redirection to create an infinite feedback loop.
Input- bash 24_redirection.sh
Output- Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling

Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling.....[infinity times]

Ques : Description-Script to use a recursive function to print each argument passed to the function.
Input-bash 25_recursion.sh 1 2 5 9 4 s jk l 7 5
Output- 1
2

5
9
4
s
jk
l
7
5

```bash
#!/bin/bash

function print_args {
    if [ $# -gt 0 ]; then
        echo $1
        shift
        print_args $@
    fi
}

print_args $@
```

```
srishtirupa@linux:~$ touch s25.sh
srishtirupa@linux:~$ chmod 777 s25.sh
srishtirupa@linux:~$ nano s25.sh
srishtirupa@linux:~$ nano s25.sh
srishtirupa@linux:~$ bash s25.sh 1 3 2 sd 6 4
1
3
2
sd
6
4
srishtirupa@linux:~$
```

Ques : Description- Script to determine whether a given file system or mount point is mounted.
Input- bash 26_mounted_fs.sh /dev/sda8
Output- File-system /dev/sda8 is mounted on / and it is having 7% used space with 227859240 KB free

```
#!/bin/bash

function find_files()
{
        var= df -h | awk '{if($5<10) print $1}'
        echo $var
}

find_files
```

Ques : Description- Script to to calculate the BMI.
Input- bash 34_BMI.sh
Enter the weight in Kg :48.2
Enter the height in meters :1.4
Output- The BMI is 25.3
You are overweight

```
#!/bin/bash


read -p "Enter the weight in Kg: " weight
read -p "Enter the height in meters: " height


bmi=$(echo "scale=1; $weight / ($height * $height)" | bc)


echo "The BMI is $bmi"


if (( $(echo "$bmi < 18.5" | bc -l) )); then
    echo "You are underweight"
elif (( $(echo "$bmi < 25" | bc -l) )); then
    echo "You have a normal weight"
elif (( $(echo "$bmi < 30" | bc -l) )); then
    echo "You are overweight"
else
    echo "You are obese"
fi
```

```
srishtirupa@linux:~$ nano s26.sh
srishtirupa@linux:~$ touch 34.sh
srishtirupa@linux:~$ chmod 777 34.sh
srishtirupa@linux:~$ nano 34.sh
srishtirupa@linux:~$ bash 34.sh
Enter the weight in Kg: 40
Enter the height in meters: 1.67
The BMI is 14.3
You are underweight
srishtirupa@linux:~$
```