

1. Write a shell script that prints "Shell Scripting is Fun!" on the screen.

Soln:

```
echo "Shell scripting is Fun!"
```

2. Modify the shell script from exercise 1 to include a variable. The variable will hold the contents of the message "Shell Scripting is Fun!".

Soln:

```
MSG="Shell scripting is Fun!"  
echo $MSG;
```

3. Write a shell script to add two numbers and print the result. The two numbers should be input by the user.

Soln:

```
echo "Enter A: "  
read A  
echo "Enter B: "  
read B  
SUM=`expr $A + $B`  
echo "Sum = $SUM "
```

4. Write a shell script to find the given number is an odd number or even number.

Soln:

```
#To find whether the num is ODD or EVEN

echo "Enter num : "

read NUM

if [ `expr $NUM % 2` == 0 ]

then
    echo "$NUM is an EVEN Number"
else
    echo "$NUM is an ODD Number"
fi
```

5. Write a shell script to print the multiplication table of a given number upto 15 times.

Soln:

```
#Multiplication Table of a number upto 15

COUNT=1

echo "Enter Number : "

read NUM

echo "\n" #new line
```

```
while [ $COUNT -lt 16 ]
do

    MUL=`expr $NUM \* $COUNT`

    echo "$NUM x $COUNT = $MUL"

    COUNT=`expr $COUNT + 1`
done
```

6. Store the output of the command “hostname” in a variable. Display “This script is running on _.” where “_” is the output of the “hostname” command.

Soln:

```
HOSTNAME=`hostname`
```

```
echo "The script is running on $HOSTNAME"
```

7. Write a shell script that displays “man”, “bear”, “pig”, “dog”, “cat”, and “sheep” on the screen with each appearing on a separate line. Try to do this in as few lines as possible.

Soln:

```
ANIMALS=("man" "bear" "pig" "dog" "cat" "sheep")

for ANIMAL in "${ANIMALS[@]}";
```

```
do
    echo $ANIMAL
done
```

8. Write a shell script that prompts the user for a name of a file or directory and reports if it is a regular file, a directory, or another type of file. Also perform an ls command against the file or directory with the long listing option.

Soln:

```
echo "Enter Directory or File-Name: "

read INPUT

if [ -f "$INPUT" -o -d "$INPUT" ] #if file or dir exists

then
    echo "`ls -l $INPUT`"
    echo "`file $INPUT`"
else
    echo "File or Directory does not exist! "
fi
```

9. Modify the previous script to that it accepts the file or directory name as an argument instead of prompting the user to enter it.

Soln:

```
# $1 is the positional argument

if [ -f "$1" -o -d "$1" ] #if file or dir exists

then
    echo "`ls -l $1`"
    echo "`file $1`"
else
    echo "File or Directory does not exist! "
fi
```

10. Write a shell script that displays the number of files in the present working directory.

Soln:

```
FILES_COUNT=`pwd | ls | wc -l`

echo "$FILES_COUNT files are in the current
directory."
```

11. Write a shell script that displays the number of files in the specified directory. The directory name should be given as input by the user.

Soln:

```
read P #read dir as input

if [ -d "$P" ]#check if the path is a valid directory
```

then

```
FILES_COUNT=`ls $P | wc -l`
```

```
echo "$FILES_COUNT files are in the given
directory."
```

else

```
echo "$P is not a valid directory!"
```

fi

12. Write the shell script that renames all files in the current directory that end in ".txt" to begin with today's date in the following format: YYYY-MM-DD. For example, if a file a.txt was in the current directory and today was March 17, 2021 it would change name from "a.txt" to "2021-03-17-a.txt".

Soln:

```
# CURRENT DATE
```

```
CURRENT_DATE=`date '+%Y-%m-%d'`
```

```
# ITERATE OVER FILES IN CURRENT DIR
```

```
for FILE in "`pwd`"/*
```

do

```
# DELTE prefix path using `basename` command
```

```
FILE="$(basename -- "$FILE")"
```

```
# extract extension
```

```
extension="${FILE##*.}"
```

```
# extract filename without extension
```

```
filename="${FILE%.*}"
```

```
#rename only txt files
```

```
if [ $extension == 'txt' ]  
  
then  
    `mv $FILE $CURRENT_DATE-$filename.txt`  
fi  
done
```