1a. #!/bin/bash

3b. #!/bin/bash

read -p "Enter directory path: " dir

if [ -d "$dir" ]; then

echo "Files in $dir:"

ls -l "$dir"

else

echo "$dir is not a directory"

fi

4.

# File: number\_datatypes.py

# Integer

int\_num = 10

print("Integer:", int\_num, type(int\_num))

# Float

float\_num = 10.5

print("Float:", float\_num, type(float\_num))

# Complex

complex\_num = 3 + 4j

print("Complex:", complex\_num, type(complex\_num))

echo "List of all files in current directory:"

ls -l

1b. #!/bin/bash

for item in "$@"

do

if [ -f "$item" ]; then

echo "$item is a file"

elif [ -d "$item" ]; then

echo "$item is a directory"

else

echo "$item is neither a file nor a directory"

fi

done

2. #!/bin/bash

for file in "$@"

do

echo "Word count in file: $file"

tr -s ' ' '\n' < "$file" | sort | uniq -c

done

3a. #!/bin/bash

5.

a = 10

b = 5

print("Addition:", a + b)

print("Subtraction:", a - b)

print("Multiplication:", a \* b)

print("Division:", a / b)

print("Modulus:", a % b)

print("Exponent:", a \*\* b)

read -p "Enter a number: " num

fact=1

for (( i=1; i<=num; i++ ))

do

fact=$((fact \* i))

done

echo "Factorial of $num is $fact"

6.

8.

# i. Create a list

set mylist [list 1 2 3]

# ii. Append elements

lappend mylist 4 5

# iii. Traverse the list

foreach item $mylist {

puts $item

}

# iv. Concatenate list

set mylist2 [list 6 7]

set newlist [concat $mylist $mylist2]

puts "Concatenated List: $newlist"

9a.

set file1 "fileA.txt"

set file2 "fileB.txt"

if {[file mtime $file1] > [file mtime $file2]} {

puts "$file1 is newer than $file2"

} else {

puts "$file2 is newer than $file1"

}

9b.

set in [open "source.txt" r]

set out [open "destination.txt" w]

fconfigure $in -translation auto

fconfigure $out -translation auto

while {[gets $in line] >= 0} {

puts $out $line

}

close $in

close $out

s1 = "Hello"

s2 = "World"

s3 = s1 + " " + s2

print("Concatenated String:", s3)

# Substring

print("Substring (0:5):", s3[0:5])

7a.

proc factorial {n} {

set result 1

for {set i 1} {$i <= $n} {incr i} {

set result [expr $result \* $i]

}

return $result

}

puts "Factorial of 5: [factorial 5]"

7b.

for {set i 1} {$i <= 10} {incr i} {

puts "Multiplication Table for $i:"

for {set j 1} {$j <= 10} {incr j} {

set result [expr {$i \* $j}]

puts "$i x $j = $result"

}

puts "-------------------------"

}