Experiment Number: 03

Problem Statement: **Write shell scripts which covers basic arithmetic, control structures, command line arguments, functions and arrays.**

NAME: **Omkar Khanvilkar**  ROLLNO: 07

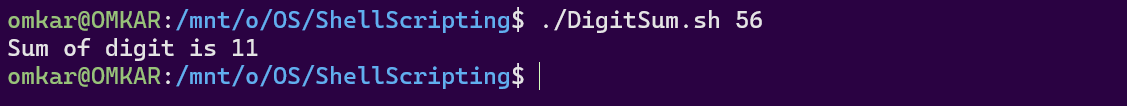
CLASS: IT/B BATCH: 2

DATE OF PERFORMANCE: 06/08/2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Print Sum of Digits of a given number using command line argument**
2. *#!/bin/bash*
3. if [ *$#* -eq 1 ]
4. then
5. num=$1
6. while [ $num -gt 0 ]
7. do
8. res=$(( $num % 10 ))
9. sum=$(( $sum + $res ))
10. num=$(( $num / 10 ))
11. done
12. echo Sum of digit is $sum
13. else
14. echo Invalide input
15. fi

**Output:**

****

**2) Write a shell script using function for following:**

1. **average of given numbers**
2. **Max  digit from given number**
3. **min digit  from given number**

*#!/bin/bash*

getAvg(){

a=*$#*

sum=0

for i in $@

do

        sum=$(( $sum + $i ))

avg=$(( $sum / $a ))

done

return $avg

}

getMax(){

        max=-2147483647

        for i in $@

        do

                if [ $i -gt $max ]

                then

                        max=$i

                fi

        done

return $max

}

getMin(){

        min=2147483647

        for i in $@

        do

                if [ $i -lt $min ]

                then

                        min=$i

                fi

        done

return $min

}

loop=1

while [ $loop -eq 1 ]

do

        echo Choose the operation from the following on given array

        echo *$\**

        echo a] Get Average

        echo b] Get Max Element

        echo c] Get Min Element

        echo d] Exit

        read choice

        case $choice in

                a) getAvg $@

                        echo Average of value : *$?*;;

                b) getMax $@

                        echo Maximum value : *$?*;;

                c) getMin $@

                        echo Minimum value : *$?*;;

                d) loop=0

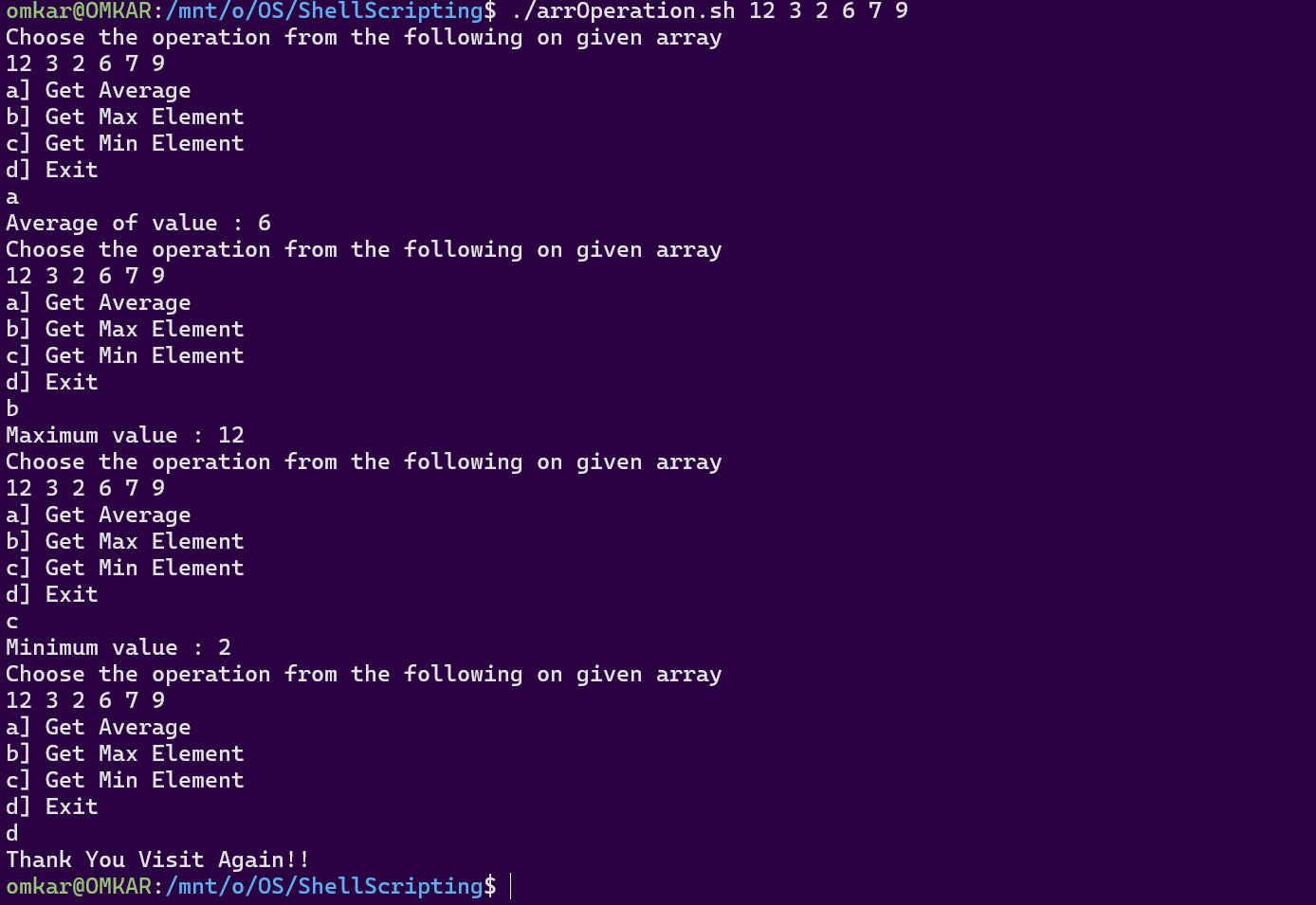
                        echo "Thank You Visit Again!!";;

                \*) echo "Invalide Input";;

        esac

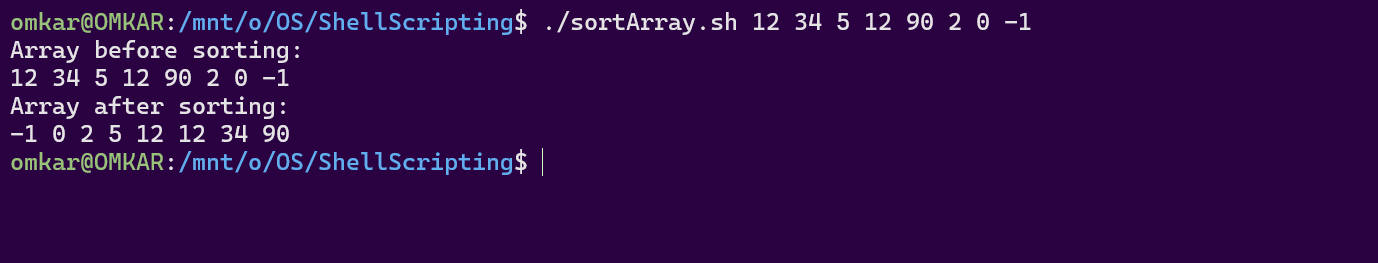
done

**Output :**

****

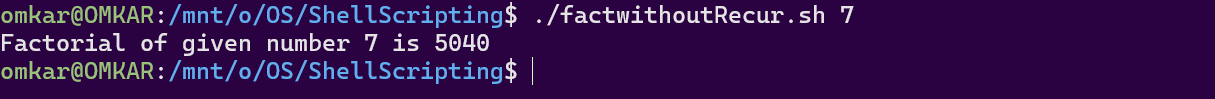
1. **Perform sorting on given array elements**
2. *#!/bin/bash*
3. *# Initialize the index and the array*
4. j=0
5. array=()
6. for i in "$@"
7. do
8. array[$j]=$i
9. let j++
10. done
11. echo "Array before sorting: "
12. echo ${array[\*]}
13. len=$(( ${#array[@]} ))
14. *# Bubble sort algorithm to sort the array*
15. for ((i=0; i<$len; i++))
16. do
17. for ((j=i+1; j<$len; j++))
18. do
19. if [ ${array[$i]} -gt ${array[$j]} ]
20. then
21. temp=${array[$i]}
22. array[$i]=${array[$j]}
23. array[$j]=$temp
24. fi
25. done
26. done
27. echo "Array after sorting: "
28. echo ${array[\*]}

**Output :**

****

1. **Program to find factorial of a given number without recursion**
2. *#!/bin/bash*
3. facto()
4. {
5. local num=$1
6. local fact=1
7. if [ $num -le 1 ]
8. then
9. return $fact
10. else
11. while [ $num -gt 0 ]
12. do
13. fact=$(( $fact *\** $num ))
14. let num--
15. done
16. echo $fact
17. fi
18. }
19. ans=$( facto $1 )
20. echo Factorial of given number $1 is $ans

**Output :**

****

**5) Program to find factorial of a given number with recursion**

*#!/bin/bash*

facto()

{

    if [ $1 -le 1 ]

    then

        echo 1

    else

        echo $(( $(facto $(( $1 - 1 )) ) \* $1 ))

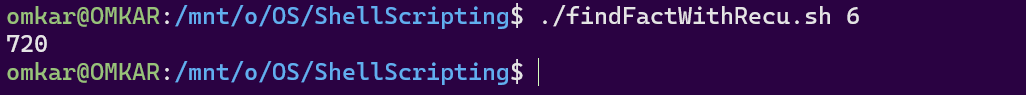
    fi

}

ans=$( facto $1 )

echo $ans

**Output :**

****

**6) Program to check file type and permission for a given file**

*#!/bin/bash*

filename=$1

echo Choose the options from following:

echo 1] Get File type

echo 2] Get File Permissions

read option

case $option in

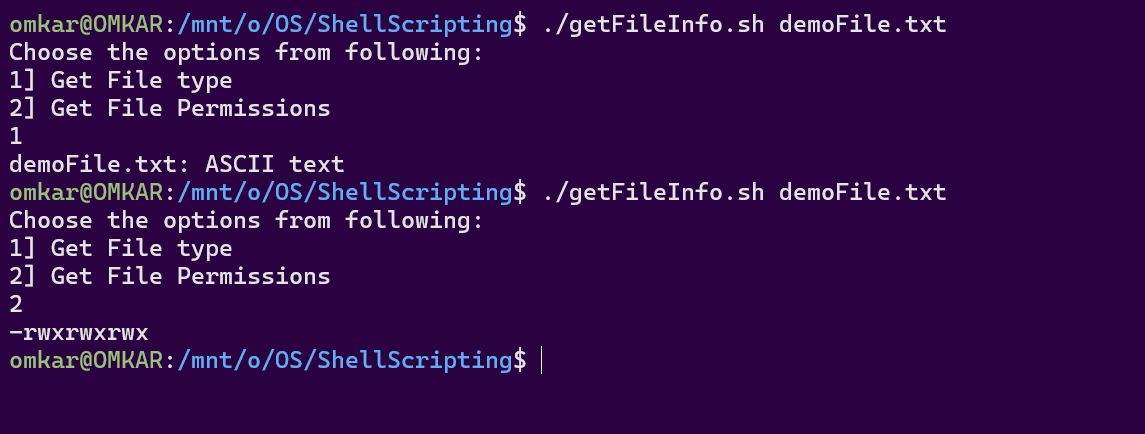
    1) echo `file $1`;;

    2) echo `stat -c %A $1`;;

*\**) echo "Invalide Input!";;

esac

**Output:**

****

**7) Check entered string is palindrome or not?**

*#!/bin/bash*

pallindrom() {

    string=$1

    revstring=$(echo "$string" | rev)

    if [[ $string == $revstring ]]; then

        return 0  *# palindrome*

    else

        return 1  *# not palindrome*

    fi

}

pallindrom "$1"

if [ *$?* -eq 0 ]; then

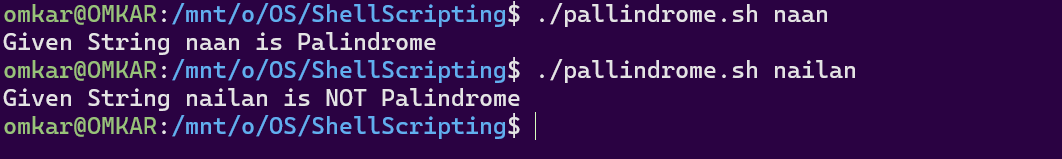
    echo "Given String $1 is Palindrome"

else

    echo "Given String $1 is NOT Palindrome"

fi

**Output:**

****