Experiment Number: 3

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

NAME: Aadesh Chawla ROLLNO: 12

CLASS: TY-IT-A BATCH: B1

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

1. **Print Sum of Digits of a given number using command line argument**

echo "Enter the number : "

read num

sum=0

while [[ $num -gt 0 ]]

do

a=`expr $num % 10`

sum=`expr $sum + $a`

num=`expr $num / 10`

done

echo "The sum of digits is : $sum"

**Output:**

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./sum.sh

Enter the number :

145

The sum of digits is : 10

1. **Write a shell script using function for following:**

* **average of given numbers**

echo "enter the total count of numbers : "

read n

for (( i=0; i < n; i++))

do

read arr[$i]

done

sum=0

for (( i=0; i < n; i++))

do

num=${arr[$i]}

sum=`expr $sum + $num`

done

avg=`expr $sum / $n`

echo "Average is $avg"

**Output:**

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./avg.sh

enter the total count of numbers :

3

1

2

3

Average is 2

* **Max from given numbers**

echo "enter the total count of numbers : "

read n

for (( i=0; i < n; i++))

do

read arr[$i]

done

max=${arr[0]}

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

do

if [ ${arr[$i]} -gt $max ]

then

max=${arr[$i]}

fi

done

echo "Maximum number is : $max"

**Output:**

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./max.sh

enter the total count of numbers :

3

56

12

678

Maximum number is : 678

* **min from given numbers**

echo "enter the total count of numbers : "

read n

for (( i=0; i < n; i++))

do

read arr[$i]

done

min=${arr[0]}

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

do

if [ ${arr[$i]} -lt $min ]

then

min=${arr[$i]}

fi

done

echo "Maximum number is : $min"

**Output:**

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./min.sh

enter the total count of numbers :

3

56

78

12

Maximum number is : 12

1. **Perform sorting on given array elements**

echo "enter the total count of numbers : "

read n

echo "enter the numbers : "

for (( i=0; i < n; i++))

do

read arr[$i]

done

for (( j=0; j < n; j++))

do

for (( i=0; i < n-j-1; i++ ))

do

if [ ${arr[$i]} -gt ${arr[$((i+1))]} ]

then

temp=${arr[$i]}

arr[$i]=${arr[$((i+1))]}

arr[$((i+1))]=$temp

fi

done

done

echo "Sorted elements : "

for (( i=0; i < n; i++))

do

echo "${arr[$i]} "

done

**Output:**

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./sort.sh

enter the total count of numbers :

3

enter the numbers :

7

6

5

Sorted elements :

5

6

7

1. **Program to find factorial of a given number with and without recursion**

echo "enter the number : "

read n

function fact\_recursion(){

if [ $1 -eq 0 ]

then

echo 1

else

a=$(($1-1))

factorial=$(fact\_recursion $a)

echo $(( $1 \* $factorial ))

fi

}

function fact\_wr(){

fact=1

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

do

fact=$(($fact\*$i))

done

echo $fact

}

fact\_recursive=$(fact\_recursion $n)

echo "Factorial of $n (with recursion): $fact\_recursive"

fact\_wr=$(fact\_wr $n)

echo "Factorial of $n (without recursion): $fact\_wr"

**Output:**

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./fact.sh

enter the number :

5

Factorial of 5 (with recursion): 120

Factorial of 5 (without recursion): 120

1. **Program to check file type and permission for a given file**

if [ $# -eq 0 ]

then

echo "ERROR : provide the filepathh as the argument"

exit 1

fi

filename=$1

if [ ! -e "$filename" ]

then

echo "ERROR : File not found"

fi

filetype=$(file -b "$filename")

permission=$(stat -c "%A" "$filename")

echo "Filetype : $filetype"

echo "Permissions : $permission"

**Output:**

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./filename.sh

ERROR : provide the filepathh as the argument

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./filename.sh test.txt

Filetype : ASCII text

Permissions : -rw-rw-r--

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./filename.sh linux.txt

ERROR : File not found

1. **Check entered string is palindrome or not?**

echo "enter a string : "

read input

reversed="${input: -1}"

for ((i=${#input} - 2; i >= 0; i--))

do

reversed="${reversed}${input:$i:1}"

done

if [ "$input" == "$reversed" ]

then

echo "String is Palindrome"

else

echo "String is not a Palindrome"

fi

**Output:**

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./palndrome.sh

enter a string :

nayan

String is Palindrome

aadesh@aadesh-VirtualBox:~/Desktop/icecream$ ./palndrome.sh

enter a string :

aadesh

String is not a Palindrome