**UNIX Lab Assignment 5**

Name: Sreekesh Iyer Roll No. 24

**Arrays in Linux**:

An array is a variable containing multiple values of the same type or of different types. There is no maximum limit to the size of an array, nor any requirement that member variables be indexed or assigned contiguously. Array index starts with zero.

1. **Reading values in an array and displaying them**: **Code**:

echo "Enter the array: " for((i=0;i<=5;i++))

do

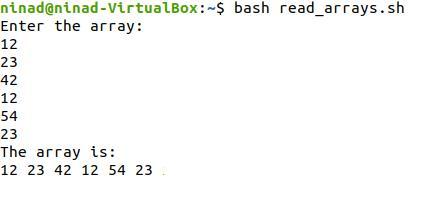
read a[$i]

done

echo "The array is: " for((i=0;i<=5;i++)) do

echo -n "${a[i]} "

done **Output**:



1. **Searching the element**: **Code**:

a=(1 2 3 4 5 6 7 8 9 10)

length=${#a[@]}

echo "The array is: "

for((i=0;i<=$length;i++))

do

echo -n "${a[i]} "

done

echo " "

echo -n "Enter the element to be searched: "

read input

j=0

fag=0

while [ $j -lt $length ]

do

if [ ${a[j]} -eq $input ]

then

fag=1

echo "Element found at $j"

break

else

j=$((j+1))

f

done

if [ $fag -eq 0 ]

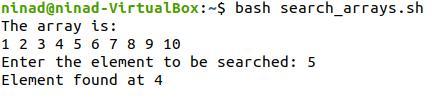
then

echo "Element is not found"

else

echo " "

**Output**:



**Functions in Linux**:

Functions enable you to break down the overall functionality of a script into smaller, logical subsections, which can then be called upon to perform their individual tasks when needed. Using functions to perform repetitive tasks is an excellent way to create code reuse. This is an important part of modern object-oriented programming principles.

Shell functions are similar to subroutines, procedures, and functions in other programming languages.

1. **Simple functions**: **Code**:

add()

{

echo $1 $2 x=$1 y=$2

return $(($x+$y))

}

read num1 num2 add $num1 $num2 echo "Sum = $?"

**Output**