**Arrays**

**1. Write a program in the following steps**

**a. Generates 10 Random 3 Digit number.**

**b. Store this random numbers into a array.**

**c. Then find the 2nd largest and the 2nd smallest element without sorting the array.**

#!/bin/bash

for((i=0;i<10;i++))

do

randomvar=$((RANDOM%899+100))

#read randomvar

array[i]=$randomvar

done

echo ${array[@]}

largest=${array[0]}

secondmaximum='unset'

for((i=1; i < ${#array[@]}; i++))

do

if [[ ${array[i]} > $largest ]]

then

secondmaximum=$largest

largest=${array[i]}

elif (( ${array[i]} != $largest )) && { [[ "$secondmaximum" = "unset" ]] || [[ ${array[i]} > $secondmaximum ]]; }

then

secondmaximum=${array[i]}

fi

done

echo "secondmaximum = $secondmaximum"

minimum=${array[0]}

secondminimum='unset'

for((i=1; i < ${#array[@]}; i++))

do

if [[ ${array[i]} < $minimum ]]

then

secondminimum=$minimum

minimum=${array[i]}

elif (( ${array[i]} != $minimum )) && { [[ "$secondminimum" = "unset" ]] || [[ ${array[i]} < $secondminimum ]]; }

then

secondminimum=${array[i]}

fi

done

echo "secondminimum = $secondminimum"

output:

AKHIL@DESKTOP-43E18QK MINGW64 ~

$ ./arrayrandom1.sh

886 221 828 330 752 431 599 458 124 864

secondmaximum = 864

secondminimum = 221

**2. Extend the above program to sort the array and then find the 2nd largest and the 2nd smallest element.**

#!/bin/bash

for((i=0;i<10;i++))

do

randomvar=$((RANDOM%899+100))

#read randomvar

arr[i]=$randomvar

done

echo ${arr[\*]}

for ((i=0;i<10;i++))

do

for ((j=0;j<9;j++))

do

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

then

# swap

temp=${arr[j]}

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

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

fi

done

done

echo "array in sorted order"

echo ${arr[\*]}

j=0

echo "${arr[$((j+1))]}" second smallest element

echo "${arr[-2]}" second largest element

output:

AKHIL@DESKTOP-43E18QK MINGW64 ~

$ ./arrayrandom2.sh

450 623 384 939 857 110 839 979 433 164

array in sorted order

110 164 384 433 450 623 839 857 939 979

164 second smallest element

939 second largest element

**3. Extend the Prime Factorization Program to store all the Prime Factors of a number n into an array and finally display the output.**

#! /bin/bash

echo "enter an integer:"

read input

if [ $input -lt 1 ];then

echo "not allowed!"

exit 1

fi

# find factors and prime

i=2

count=0

flag=0

for ((i;i<$input;));do

if [ `expr $input % $i` -eq 0 ];then

factor=$i

for ((j=2;j<=`expr $factor / 2`;));do

flag=0

if [ `expr $factor % $j` -eq 0 ];then

flag=1

break

fi

j=`expr $j + 1`

done

if [ $flag -eq 0 ];then

echo "[ $factor ]"

count=1

fi

fi

i=`expr $i + 1`

done

if [ $count -eq 0 ];then

echo "no prime factors found except 1 and $input"

arr=( 1 $input)

for i in arr

do

echo ${arr[@]}

done

fi

output:

$ ./primearray.sh

enter an integer:

23

no prime factors found except 1 and 23

1 23

**4. Sum of the three integer adds to zero**

#! /bin/bash

arr=(1 3 -4)

echo ${arr[@]}

sum=0

for i in ${arr[@]}

do

sum=`expr $sum + $i`

done

echo $sum

output:

AKHIL@DESKTOP-43E18QK MINGW64 ~/TerminalCommands/linux-content (master)

$ ./sumarray.sh

1 3 -4

0

**5. Take a range from 0 – 100, find the digits that are repeated twice like 33, 77, etc and store them in an array**

#! /bin/bash -x

position=0

for ((i=1;i<=100;i++))

do

var=$((i%11))

if [ $var -eq 0 ]

then

((array[position]=$i))

((position++))

fi

done

echo ${array[@]}

output:

AKHIL@DESKTOP-43E18QK MINGW64 ~

AKHIL@DESKTOP-43E18QK MINGW64 ~

$ ./array6.sh

11 22 33 44 55 66 77 88 99