ARRAY BASED PROBLEMS

Q1.> 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.
Ans >>> program for the problem.

```
#!/bin/bash -x
size=10;
index=0;
declare -a array;
while [ $index -lt  $size ]
do
  array[$index]=$(( (RANDOM%100) + 100 )) ;
 ((index++));
done
   echo "${array[@]}"

largest=${array[0]}
secondGreatest='unset'

for((i=1; i < ${#array[@]}; i++))
do
  if [[ ${array[i]} > $largest ]]
  then
      secondGreatest=$largest
      largest=${array[i]}
  elif (( ${array[i]} != $largest )) && { [[ "$secondGreatest" = "unset"
]] || [[ ${array[i]} > $secondGreatest ]]; }
  then
      secondGreatest=${array[i]}
  fi
done
echo "secondGreatest = $secondGreatest"

smallest=${array[0]}
secondSmallest='unset'

for((i=1; i < ${#array[@]}; i++))
do
   if [[ ${array[i]}< $smallest ]]
  then
      secondSmallest=$smallest
  elif (( ${array[i]} != $smallest )) && { [[ "$secondSmallest" = "unset"
]] || [[ ${array[i]} < $secondSmallest ]]; }
  then
      secondSmallest=${array[i]}
  fi
done
echo "secondSmallest = $secondSmallest"
```

Q2.>Extend the above program to sort the array and then find the 2nd largest
and the 2nd smallest element

Ans >> program for the problem

```
#!/bin/bash
size=10;
```

```
index=0;
declare -a array;
while [ $index -lt  $size ]
do
  array[$index]=$(( (RANDOM%100) + 100 )) ;
 ((index++));
done
   echo "${array[@]}"
secondGreatest=$(printf '%s\n' "${array[@]}" | sort -nu | tail -2 | head
-1 )
secondSmallest=$(printf '%s\n' "${array[@]}" | sort -n | head -2 | tail -
1 )
echo "SecondGreatest value is : "  $secondGreatest
echo "SecondSmallestest value is : "  $secondSmallest
```

Q3.>Write a Program to show Sum of three Integer adds to ZERO

Ans.>> program for the problem

```
#!/bin/bash -x
arr=(1 -2 1 3 -2 -5)
n=${#arr[@]};
for (( i=0;i<$n-3;i++ ))
do
for (( j=i+1;j<$n-2;j++ ))
do
for (( k=j+1;k<$n-1;k++))
do
   if  (( arr[i] + arr[j] + arr[k] == 0 ))
   then
      echo "sum of integers : " ${arr[i]} ${arr[j]} ${arr[k]} " is
triplet";
   else
       continue
fi
done
done
done
```

Q4.> Take a range from 0 - 100, find the digits that are repeated twice like 33, 77,
etc and store them in an array

Ans.>> program for the problem

```
#!/bin/bash -x
k=0;
for(( i=10; i<=99; i++ ))
do
num=$i
while [ $num -gt 0 ]
do
      rev=0
      rem=$(($num%10))
      rev=$(($rev*10+$rem))
      num=$(($num/10))
if [ $num -eq $rev ]
then
      arr[$((k++))]=$i
fi
done
```

```
done

for(( j=0; j<k; j++ ))
do
      echo ${arr[j]}
done
```

Q5.> Extend the Prime Factorization Program to store all the Prime Factors of a
number n into an array and finally display the output.

Ans .>>>> program for prime factorization

```
#!/bin/bash -x
flag=0;

function addPrimeNumber()
 {
   arr=$1
  for (( i=2; $i<=$num; i++ ))
   do
   if [ $(($num%$i)) -eq 0 ]
   then
      arr[flag]=$i
      ((flag++))
      num=$(($num/$i))
      ((i--))
   fi
  done
        echo ${arr[@]}
}

read -p "Enter a number:" num

echo "Array :" "$( addPrimeNumber $((num)))"
```

Array problems complete here.