**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.**

**Solution-**

#!/bin/bash -x

smallest=1000

secondSmallest=0

highest=0

secondHighest=0

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

do

random=$((100+RANDOM%999))

array[((i))]=$random

done

for i in "${array[@]}"

do

if [[ $i -gt $highest ]]

then

secondHighest=$highest

highest=$i

elif [[ $i -gt $secondHighest ]]

then

secondHighest=$i

fi

if [[ $i -lt $smallest ]]

then

secondSmallest=$smallest

smallest=$i

elif [[ $i -lt $secondSmallest ]]

then

secondSmallest=$i

fi

done

echo $secondHighest

echo $secondSmallest

**Output-**

$ ./2ndMinMax.sh

+ smallest=1000

+ secondSmallest=0

+ highest=0

+ secondHighest=0

+ (( i=0 ))

+ (( i<10 ))

+ random=163

+ array[((i))]=163

+ (( i++ ))

+ (( i<10 ))

+ random=750

+ array[((i))]=750

+ (( i++ ))

+ (( i<10 ))

+ random=179

+ array[((i))]=179

+ (( i++ ))

+ (( i<10 ))

+ random=155

+ array[((i))]=155

+ (( i++ ))

+ (( i<10 ))

+ random=602

+ array[((i))]=602

+ (( i++ ))

+ (( i<10 ))

+ random=488

+ array[((i))]=488

+ (( i++ ))

+ (( i<10 ))

+ random=312

+ array[((i))]=312

+ (( i++ ))

+ (( i<10 ))

+ random=754

+ array[((i))]=754

+ (( i++ ))

+ (( i<10 ))

+ random=486

+ array[((i))]=486

+ (( i++ ))

+ (( i<10 ))

+ random=496

+ array[((i))]=496

+ (( i++ ))

+ (( i<10 ))

+ for i in "${array[@]}"

+ [[ 163 -gt 0 ]]

+ secondHighest=0

+ highest=163

+ [[ 163 -lt 1000 ]]

+ secondSmallest=1000

+ smallest=163

+ for i in "${array[@]}"

+ [[ 750 -gt 163 ]]

+ secondHighest=163

+ highest=750

+ [[ 750 -lt 163 ]]

+ [[ 750 -lt 1000 ]]

+ secondSmallest=750

+ for i in "${array[@]}"

+ [[ 179 -gt 750 ]]

+ [[ 179 -gt 163 ]]

+ secondHighest=179

+ [[ 179 -lt 163 ]]

+ [[ 179 -lt 750 ]]

+ secondSmallest=179

+ for i in "${array[@]}"

+ [[ 155 -gt 750 ]]

+ [[ 155 -gt 179 ]]

+ [[ 155 -lt 163 ]]

+ secondSmallest=163

+ smallest=155

+ for i in "${array[@]}"

+ [[ 602 -gt 750 ]]

+ [[ 602 -gt 179 ]]

+ secondHighest=602

+ [[ 602 -lt 155 ]]

+ [[ 602 -lt 163 ]]

+ for i in "${array[@]}"

+ [[ 488 -gt 750 ]]

+ [[ 488 -gt 602 ]]

+ [[ 488 -lt 155 ]]

+ [[ 488 -lt 163 ]]

+ for i in "${array[@]}"

+ [[ 312 -gt 750 ]]

+ [[ 312 -gt 602 ]]

+ [[ 312 -lt 155 ]]

+ [[ 312 -lt 163 ]]

+ for i in "${array[@]}"

+ [[ 754 -gt 750 ]]

+ secondHighest=750

+ highest=754

+ [[ 754 -lt 155 ]]

+ [[ 754 -lt 163 ]]

+ for i in "${array[@]}"

+ [[ 486 -gt 754 ]]

+ [[ 486 -gt 750 ]]

+ [[ 486 -lt 155 ]]

+ [[ 486 -lt 163 ]]

+ for i in "${array[@]}"

+ [[ 496 -gt 754 ]]

+ [[ 496 -gt 750 ]]

+ [[ 496 -lt 155 ]]

+ [[ 496 -lt 163 ]]

+ echo 750

750

+ echo 163

163

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

**Solution-**

#!/bin/bash -x

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

do

random=$((100+RANDOM%999))

array[((i))]=$random

done

echo ${array[@]}

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

do

flag=0

for ((j=0;j<9-$i;j++))

do

if [[ ${array[j]} -gt ${array[j+1]} ]]

then

temp=${array[j]}

array[j]=${array[j+1]}

array[j+1]=$temp

flag=1

fi

done

if [[ $flag -eq 0 ]]

then

break;

fi

done

echo "${array[@]}"

echo "${array[1]}"

echo "${array[8]}"

**Output-**

.

.

.

+ (( j++ ))

+ (( j<9-7 ))

+ [[ 308 -gt 261 ]]

+ temp=308

+ array[j]=261

+ array[j+1]=308

+ flag=1

+ (( j++ ))

+ (( j<9-7 ))

+ [[ 1 -eq 0 ]]

+ (( i++ ))

+ (( i<10 ))

+ flag=0

+ (( j=0 ))

+ (( j<9-8 ))

+ [[ 195 -gt 261 ]]

+ (( j++ ))

+ (( j<9-8 ))

+ [[ 0 -eq 0 ]]

+ break

+ echo 195 261 308 395 416 455 484 648 720 948

195 261 308 395 416 455 484 648 720 948

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

**Solution-**

#!/bin/bash -x

read -p "Enter the first number : " number

prime=$number

index=0

for (( i=2;i<$number/2;i++ ))

do

count=0

for (( j=2;j<$i;j++ ))

do

if [[ $i%$j -eq 0 ]]

then

count=1;

break;

fi

done

if [[ count -eq 0 ]]

then

while [[ $prime%$i -eq 0 ]]

do

array[((index))]=$i

prime=$(($prime/$i))

((index++))

done

fi

done

echo ${array[@]}

**Output-**

$ ./primeFactorsArray.sh

+ read -p 'Enter the first number : ' number

Enter the first number : 12

+ prime=12

+ index=0

+ (( i=2 ))

+ (( i<12/2 ))

+ count=0

+ (( j=2 ))

+ (( j<2 ))

+ [[ count -eq 0 ]]

+ [[ 12%2 -eq 0 ]]

+ array[((index))]=2

+ prime=6

+ (( index++ ))

+ [[ 6%2 -eq 0 ]]

+ array[((index))]=2

+ prime=3

+ (( index++ ))

+ [[ 3%2 -eq 0 ]]

+ (( i++ ))

+ (( i<12/2 ))

+ count=0

+ (( j=2 ))

+ (( j<3 ))

+ [[ 3%2 -eq 0 ]]

+ (( j++ ))

+ (( j<3 ))

+ [[ count -eq 0 ]]

+ [[ 3%3 -eq 0 ]]

+ array[((index))]=3

+ prime=1

+ (( index++ ))

+ [[ 1%3 -eq 0 ]]

+ (( i++ ))

+ (( i<12/2 ))

+ count=0

+ (( j=2 ))

+ (( j<4 ))

+ [[ 4%2 -eq 0 ]]

+ count=1

+ break

+ [[ count -eq 0 ]]

+ (( i++ ))

+ (( i<12/2 ))

+ count=0

+ (( j=2 ))

+ (( j<5 ))

+ [[ 5%2 -eq 0 ]]

+ (( j++ ))

+ (( j<5 ))

+ [[ 5%3 -eq 0 ]]

+ (( j++ ))

+ (( j<5 ))

+ [[ 5%4 -eq 0 ]]

+ (( j++ ))

+ (( j<5 ))

+ [[ count -eq 0 ]]

+ [[ 1%5 -eq 0 ]]

+ (( i++ ))

+ (( i<12/2 ))

+ echo 2 2 3

2 2 3

**4] Write a Program to show Sum of three Integer adds to ZERO**

**Solution-**

#!/bin/bash -x

sum=0

for ((i=1;i<4;i++))

do

read -p "Enter the numbers $i: " number

array[((i))]=$number

sum=$(($sum+${array[i]}))

done

echo ${array[@]}

echo $sum

if [[ $sum -eq 0 ]]

then

echo sum is equal to zero

else

echo sum is not equal to zero

fi

Output-

$ ./sumToZero.sh

+ sum=0

+ (( i=1 ))

+ (( i<4 ))

+ read -p 'Enter the numbers 1: ' number

Enter the numbers 1: 6

+ array[((i))]=6

+ sum=6

+ (( i++ ))

+ (( i<4 ))

+ read -p 'Enter the numbers 2: ' number

Enter the numbers 2: -3

+ array[((i))]=-3

+ sum=3

+ (( i++ ))

+ (( i<4 ))

+ read -p 'Enter the numbers 3: ' number

Enter the numbers 3: -3

+ array[((i))]=-3

+ sum=0

+ (( i++ ))

+ (( i<4 ))

+ echo 6 -3 -3

6 -3 -3

+ echo 0

0

+ [[ 0 -eq 0 ]]

+ echo sum is equal to zero

sum is equal to zero

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

**Solution-**

#!/bin/bash -x

index=0

for ((i=0;i<=100;i++))

do

num=$i

reverse=0

while [[ $num -gt 0 ]]

do

add=$num%10

reverse=$((($reverse\*10)+$add))

num=$(($num/10))

done

if [[ $i -eq $reverse && ${#i} -eq 2 ]]

then

array[((index))]=$i

((index++))

fi

done

echo ${array[@]}

**Output-**

.

.

.

+ (( index++ ))

+ (( i++ ))

+ (( i<=100 ))

+ num=100

+ reverse=0

+ [[ 100 -gt 0 ]]

+ add=100%10

+ reverse=0

+ num=10

+ [[ 10 -gt 0 ]]

+ add=10%10

+ reverse=0

+ num=1

+ [[ 1 -gt 0 ]]

+ add=1%10

+ reverse=1

+ num=0

+ [[ 0 -gt 0 ]]

+ [[ 100 -eq 1 ]]

+ (( i++ ))

+ (( i<=100 ))

+ echo 11 22 33 44 55 66 77 88 99

11 22 33 44 55 66 77 88 99