ASSIGNMENT DAY7

Problem 1] Write a program in the following steps:

- **a) Generates 10 Random three digit number**
- **b) Store this random number in an array**
- **c) Then find the 2nd largest and 2nd smallest element without sorting the array.**

Solution: nano arr1.sh

```
#!/bin/bash

for ((count=0; count<10; count++))
do
        n=$((100+RANDOM%900))
        echo "$n"
        arr[$count]=$((n))
done

echo "Elements in the array are:"${arr[@]}

small=${arr[0]}
large=${arr[0]}

for (( count=0; count<10; count++ ))
do
        if [ ${arr[$count]} -lt $small ]
        then
                small=${arr[$count]}
        elif [ ${arr[$count]} -gt $large ]
        then
                large=${arr[$count]}
        fi
done

secsmall=${arr[0]}
seclarge=${arr[0]}

for ((count=0; count<10; count++))
do
        if [ ${arr[$count]} -gt $small ] && [ ${arr[$count]} -
lt $secsmall ]
        then
                secsmall=${arr[$count]}
        elif [ ${arr[$count]} -lt $large ] && [ ${arr[$count]}
-gt $seclarge ]
        then
                seclarge=${arr[$count]}
        fi
done
echo "Second smallest number in the array is $secsmall"
echo "Second largest number in the array is $seclarge"
```

Output: chmod +x arr1.sh

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
```

```
$  ./arr1.sh
126
741
717
290
920
816
140
970
150
691
Elements in the array are:126 741 717 290 920 816 140 970 150
691
Second smallest number in the array is 140
Second largest number in the array is 920
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./arr1.sh
467
176
974
318
775
942
285
502
586
792
Elements in the array are:467 176 974 318 775 942 285 502 586
792
Second smallest number in the array is 285
Second largest number in the array is 942
```

Problem 2] Extend the above program to sort the array and then find the second largest and second smallest number.

Solution: nano arr2.sh

```
#!/bin/bash
for (( count=0; count<10; count++ ))
do
   n=$((100+RANDOM%900))
   arr[$count]=$((n))
done
echo "Elements in the array before sorting are:"${arr[@]}

temp=0
for (( count=0; count<10; count++ ))
do
   for (( c=count+1; c<10; c++ ))
   do
      if [ ${arr[count]} -gt ${arr[$((c))]} ]
      then
            temp=${arr[count]}
            arr[$count]=${arr[$((c))]}
            arr[$((c))]=$temp
      fi
   done
done
echo "Elements in the array after sorting are: ${arr[@]}"
echo "Second smallest element is: ${arr[1]}"
echo "Second largest element is: ${arr[8]}"
```

Output: chmod +x arr2.sh

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./arr2.sh
Elements in the array before sorting are:167 773 957 252 161
868 937 657 880 820
Elements in the array after sorting are: 161 167 252 657 773
820 868 880 937 957
Second smallest element is: 167
Second largest element is: 937
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./arr2.sh
Elements in the array before sorting are:547 572 829 687 893
402 775 682 695 126
Elements in the array after sorting are: 126 402 547 572 682
687 695 775 829 893
Second smallest element is: 402
Second largest element is: 829
```

Problem 3] Extend the Prime Factorization program to store the Prime factors of a number n into an array and finally display its output.

Solution: nano arr3.sh

```
#!/bin/bash
echo "Enter the number:"
read n
echo "Prime factors of $n are:"
for (( count=2; count<=n; count++ ))
do
        while [ $((n%count)) -eq 0 ]
   do
                echo $count
      n=$((n/$count))
                arr[$count]=$((count))
        done
done
echo "Array of prime factors are:" ${arr[@]}
```

Output: chmod +x arr3.sh

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./arr3.sh
Enter the number:
45
Prime factors of 45 are:
3
3
```

```
5
Array of prime factors are: 3 5

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./arr3.sh
Enter the number:
89
Prime factors of 89 are:
89
Array of prime factors are: 89

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./arr3.sh
Enter the number:
90
Prime factors of 90 are:
2
3
3
5
Array of prime factors are: 2 3 5
```

Problem 4] Write a program to show sum of three Integer adds to zero.

Solution: nano arr4.sh

```
#!/bin/bash
echo "Enter the total numbers: "
read l
echo "Enter numbers: "
i=0
while [ $i -lt $l ]
do
   read arr[$i]
   i=`expr $i + 1`
done
echo "${arr[@]}"
for ((count=0;count<=l;count++))
do
   sum=$((sum+arr[count]))
done
echo "sum: $sum"
y=$sum
if [ $y -eq 0 ]
then
   echo "Sum of array ${arr[@]} is $sum"
else
   echo "Array does not contains the elements whose sum can be zero"
fi
```

Output: chmod +x arr4.sh

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./arr4.sh
Enter the total numbers:
3
Enter numbers:
1
2
-3
1 2 -3
sum: 0
Sum of array 1 2 -3 is 0

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./arr4.sh
```

```
Enter the total numbers:
4
Enter numbers:
2
4
1
3
2 4 1 3
sum: 10
Array does not contains the elements whose sum can be zero
```

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

Solution: nano arr5.sh

```
#!/bin/bash
for ((count=1; count<=100; count++))
do
   x=$((count%11))
   if [ $x -eq 0 ]
   then
         arr[count]=$x
   fi
done
echo "Array containing repeated digits from 0 to 100 are:
${!arr[@]}"
```

Output: chmod +x arr5.sh

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./arr5.sh
Array containing repeated digits from 0 to 100 are: 11 22 33
44 55 66 77 88 99
```