Ques 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 :

```
$ cat Q1.sh
#!/bin/bash -x
count=0
for a in $(seq 10)
do
        randomNum=$(( RANDOM % 900 + 100 ))
        arr[((count++))]=$randomNum
done
echo ${arr[@]}
max=${arr[0]}
min=${arr[0]}
max2=${arr[0]}
min2=${arr[0]}
for i in "${arr[@]}"
do
    if [[ "$i" -gt "$max" ]]; then
        max2="$max"
        max="$i"
    fi
    if [[ "$i" -lt "$min" ]]; then
        min2="$min"
        min="$i"
    fi
done
echo "Array : "${arr[@]}
echo "2nd Largest Number : $max2"
echo "2nd Smallest Number : $min2"
```

```
$ ./Q1.sh
+ count=0
++ seq 10
+ for a in $(seq 10)
+ randomNum=436
+ arr[((count++))]=436
+ for a in $(seq 10)
+ randomNum=418
+ arr[((count++))]=418
+ for a in $(seq 10)
+ randomNum=852
+ arr[((count++))]=852
+ for a in $(seq 10)
+ randomNum=754
+ arr[((count++))]=754
+ for a in $(seq 10)
+ randomNum=296
+ arr[((count++))]=296
+ for a in $(seq 10)
+ randomNum=689
+ arr[((count++))]=689
+ for a in $(seq 10)
+ randomNum=340
+ arr[((count++))]=340
+ for a in $(seq 10)
+ randomNum=620
+ arr[((count++))]=620
+ for a in $(seq 10)
+ randomNum=202
+ arr[((count++))]=202
+ for a in $(seq 10)
+ randomNum=936
+ arr[((count++))]=936
+ echo 436 418 852 754 296 689 340 620 202 936
436 418 852 754 296 689 340 620 202 936
+ max=436
+ min=436
+ max2=436
+ min2=436
+ for i in "${arr[@]}"
+ [[ 436 -gt 436 ]]
+ [[ 436 -lt 436 ]]
+ for i in "${arr[@]}"
+ [[ 418 -gt 436 ]]
+ [[ 418 -lt 436 ]]
+ min2=436
+ min=418
+ for i in "${arr[@]}"
+ [[ 852 -gt 436 ]]
+ max2=436
+ max=852
+ [[ 852 -lt 418 ]]
+ for i in "${arr[@]}"
+ [[ 754 -gt 852 ]]
+ [[ 754 -lt 418 ]]
+ for i in "${arr[@]}"
+ [[ 296 -gt 852 ]]
+ [[ 296 -lt 418 ]]
+ min2=418
+ min=296
+ for i in "${arr[@]}"
+ [[ 689 -gt 852 ]]
+ [[ 689 -lt 296 ]]
+ for i in "${arr[@]}"
+ [[ 340 -gt 852 ]]
+ [[ 340 -lt 296 ]]
+ for i in "${arr[@]}"
+ [[ 620 -gt 852 ]]
+ [[ 620 -lt 296 ]]
+ for i in "${arr[@]}"
+ [[ 202 -gt 852 ]]
+ [[ 202 -lt 296 ]]
+ min2=296
+ min=202
+ for i in "${arr[@]}"
+ [[ 936 -gt 852 ]]
+ max2=852
+ max=936
+ [[ 936 -lt 202 ]]
+ echo 'Array : 436' 418 852 754 296 689 340 620 202 936
Array : 436 418 852 754 296 689 340 620 202 936
+ echo '2nd Largest Number : 852'
2nd Largest Number : 852
+ echo '2nd Smallest Number : 296'
2nd Smallest Number : 296
```

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

Solution :

```
$ cat Q2.sh
#!/bin/bash -x
```

```
arr=()
for a in $(seq 10)
do
	randomNum=$(( RANDOM % 900 + 100 ))
	arr[$a]=$randomNum
done
echo ${arr[@]}
secondLargest=$(printf '%s\n' "${arr[@]}" | sort -n | tail -2 | head -1)
secondSmallest=$(printf '%s\n' "${arr[@]}" | sort -n | head -2 | tail -1)
echo "Second Largest : "$secondLargest
echo "Second Smallest : "$secondSmallest
```

$ ./Q2.sh

```
+ arr=()
++ seq 10
+ for a in $(seq 10)
+ randomNum=638
+ arr[$a]=638
+ for a in $(seq 10)
+ randomNum=232
+ arr[$a]=232
+ for a in $(seq 10)
+ randomNum=523
+ arr[$a]=523
+ for a in $(seq 10)
+ randomNum=948
+ arr[$a]=948
+ for a in $(seq 10)
+ randomNum=953
+ arr[$a]=953
+ for a in $(seq 10)
+ randomNum=389
+ arr[$a]=389
+ for a in $(seq 10)
+ randomNum=652
+ arr[$a]=652
+ for a in $(seq 10)
+ randomNum=178
+ arr[$a]=178
+ for a in $(seq 10)
+ randomNum=468
+ arr[$a]=468
+ for a in $(seq 10)
+ randomNum=247
+ arr[$a]=247
+ echo 638 232 523 948 953 389 652 178 468 247
638 232 523 948 953 389 652 178 468 247
++ printf '%s\n' 638 232 523 948 953 389 652 178 468 247
++ sort -n
++ tail -2
++ head -1
+ secondLargest=948
++ printf '%s\n' 638 232 523 948 953 389 652 178 468 247
++ sort -n
++ head -2
++ tail -1
+ secondSmallest=232
+ echo 'Second Largest : 948'
Second Largest : 948
+ echo 'Second Smallest : 232'
Second Smallest : 232
```

Ques 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 :

```
$ cat Q3.sh
#!/bin/bash -x
read -p "Enter any number : " n
j=0
for (( i=2; i<=n; ))
do
    if [ $(($n%$i)) -eq 0 ]
    then
        n=$(($n/$i))
        arr[j++]=$i
        continue
    fi
    i=$(($i+1))
done
echo "Prime Factors in Array : ${arr[@]}"
```

```
$ ./Q3.sh
+ read -p 'Enter any number : ' n
Enter any number : 10
```

```
+ j=0
+ (( i=2 ))
+ (( i<=n ))
+ '[' 0 -eq 0 ']'
+ n=5
+ arr[j++]=2
+ continue
+ (( 1 ))
+ (( i<=n ))
+ '[' 1 -eq 0 ']'
+ i=3
+ (( 1 ))
+ (( i<=n ))
+ '[' 2 -eq 0 ']'
+ i=4
+ (( 1 ))
+ (( i<=n ))
+ '[' 1 -eq 0 ']'
+ i=5
+ (( 1 ))
+ (( i<=n ))
+ '[' 0 -eq 0 ']'
+ n=1
+ arr[j++]=5
+ continue
+ (( 1 ))
+ (( i<=n ))
+ echo 'Prime Factors in Array : 2' 5
Prime Factors in Array : 2 5
```

Ques 4. Write a program to show sum of three integer add to ZERO.

Solution :

```
$ cat Q4.sh
#!/bin/bash -x
read -p "Enter the size of an array : " size
found=false;
for (( count=0; count<$size; count++ ))
do
	read -p "Enter the arr[$count] : " value
	arr[count]=$value
done
```

```
echo  "Array : [ ${arr[@]} ]"
for (( i=0; i<`expr $size-2`; i++ ))
do
        for (( j=`expr $i+1`; j<`expr $size-1`; j++ ))
        do
                for (( k=`expr $j+1`; k<$size; k++ ))
                do
                        num1=${arr[i]}
                        num2=${arr[j]}
                        num3=${arr[k]}
                        sum=`expr $num1 + $num2 + $num3`
                        if [[ $sum==0 ]]
                        then
                                echo "$num1 : $num2 : $num3"
                                found=true;
                                echo "Triplets Exist"
                        fi
                done
        done
done
if [ $found=="false" ]
then
        echo "Triplets Not Exist"
fi
```

```
$ ./Q4.sh
+ read -p 'Enter the size of an array : ' size
Enter the size of an array : 3
+ (( count=0 ))
+ (( count<3 ))
+ read -p 'Enter the arr[0] : ' value
Enter the arr[0] : 1
+ arr[count]=1
+ (( count++  ))
+ (( count<3 ))
+ read -p 'Enter the arr[1] : ' value
Enter the arr[1] : 0
+ arr[count]=0
+ (( count++  ))
+ (( count<3 ))
+ read -p 'Enter the arr[2] : ' value
Enter the arr[2] : -1
+ arr[count]=-1
+ (( count++  ))
+ (( count<3 ))
+ echo 'Array : [ 1' 0 '-1 ]'
Array : [ 1 0 -1 ]
+ (( i=0 ))
++ expr 3-2
+ (( i<3-2 ))
++ expr 0+1
+ (( j=0+1 ))
++ expr 3-1
+ (( j<3-1 ))
++ expr 1+1
+ (( k=1+1 ))
+ (( k<3 ))
+ num1=1
+ num2=0
+ num3=-1
++ expr 1 + 0 + -1
+ sum=0
+ [[ -n 0==0 ]]
+ echo '1 : 0 : -1'
1 : 0 : -1
+ echo 'Triplets Exist'
Triplets Exist
```

Ques 5.Take a range from 0-100, find the digit that are repeated twice like 33,77, etc. and store them in an array.
Solution :

```
$ cat Q5.sh
#!/bin/bash -x
arr=()
for a in $(seq 100)
do
    echo $a
    b=$((a%11))
    if [ $b -eq 0 ]
    then
        arr[$a]=$a
    fi
done
echo ${arr[@]}
```

```
$ ./Q5.sh
+ arr=()
++ seq 100
+ for a in $(seq 100)
+ echo 1
1
+ b=1
+ '[' 1 -eq 0 ']'
+ for a in $(seq 100)
+ echo 2
2
+ b=2
+ '[' 2 -eq 0 ']'
+ for a in $(seq 100)
+ echo 3
3
+ b=3
+ '[' 3 -eq 0 ']'
+ for a in $(seq 100)
+ echo 4
4
+ b=4
+ '[' 4 -eq 0 ']'
+ for a in $(seq 100)
+ echo 5
5
+ b=5
+ '[' 5 -eq 0 ']'
+ for a in $(seq 100)
+ echo 6
6
+ b=6
+ '[' 6 -eq 0 ']'
+ for a in $(seq 100)
+ echo 7
7
+ b=7
+ '[' 7 -eq 0 ']'
+ for a in $(seq 100)
+ echo 8
8
+ b=8
+ '[' 8 -eq 0 ']'
+ for a in $(seq 100)
+ echo 9
9
+ b=9
+ '[' 9 -eq 0 ']'
+ for a in $(seq 100)
+ echo 10
10
+ b=10
+ '[' 10 -eq 0 ']'
+ for a in $(seq 100)
+ echo 11
11
+ b=0
+ '[' 0 -eq 0 ']'
+ arr[$a]=11
+ for a in $(seq 100)
+ echo 12
12
+ b=1
+ '[' 1 -eq 0 ']'
+ for a in $(seq 100)
+ echo 13
13
+ b=2
+ '[' 2 -eq 0 ']'
+ for a in $(seq 100)
+ echo 14
14
+ b=3
+ '[' 3 -eq 0 ']'
+ for a in $(seq 100)
+ echo 15
15
+ b=4
+ '[' 4 -eq 0 ']'
+ for a in $(seq 100)
+ echo 16
16
+ b=5
+ '[' 5 -eq 0 ']'
+ for a in $(seq 100)
+ echo 17
17
+ b=6
+ '[' 6 -eq 0 ']'
+ for a in $(seq 100)
+ echo 18
18
+ b=7
+ '[' 7 -eq 0 ']'
+ for a in $(seq 100)
+ echo 19
19
+ b=8
+ '[' 8 -eq 0 ']'
+ for a in $(seq 100)
+ echo 20
20
+ b=9
+ '[' 9 -eq 0 ']'
+ for a in $(seq 100)
+ echo 21
21
+ b=10
+ '[' 10 -eq 0 ']'
+ for a in $(seq 100)
+ echo 22
22
+ b=0
+ '[' 0 -eq 0 ']'
+ arr[$a]=22
+ for a in $(seq 100)
+ echo 23
23
+ b=1
+ '[' 1 -eq 0 ']'
+ for a in $(seq 100)
+ echo 24
24
+ b=2
+ '[' 2 -eq 0 ']'
+ for a in $(seq 100)
+ echo 25
25
```

```
+ b=3
+ '[' 3 -eq 0 ']'
+ for a in $(seq 100)
+ echo 26
26
+ b=4
+ '[' 4 -eq 0 ']'
+ for a in $(seq 100)
+ echo 27
27
+ b=5
+ '[' 5 -eq 0 ']'
+ for a in $(seq 100)
+ echo 28
28
+ b=6
+ '[' 6 -eq 0 ']'
+ for a in $(seq 100)
+ echo 29
29
+ b=7
+ '[' 7 -eq 0 ']'
+ for a in $(seq 100)
+ echo 30
30
+ b=8
+ '[' 8 -eq 0 ']'
+ for a in $(seq 100)
+ echo 31
31
+ b=9
+ '[' 9 -eq 0 ']'
+ for a in $(seq 100)
+ echo 32
32
+ b=10
+ '[' 10 -eq 0 ']'
+ for a in $(seq 100)
+ echo 33
33
+ b=0
+ '[' 0 -eq 0 ']'
+ arr[$a]=33
+ for a in $(seq 100)
+ echo 34
34
+ b=1
+ '[' 1 -eq 0 ']'
+ for a in $(seq 100)
+ echo 35
35
+ b=2
+ '[' 2 -eq 0 ']'
+ for a in $(seq 100)
+ echo 36
36
+ b=3
+ '[' 3 -eq 0 ']'
+ for a in $(seq 100)
+ echo 37
37
+ b=4
+ '[' 4 -eq 0 ']'
+ for a in $(seq 100)
+ echo 38
38
+ b=5
+ '[' 5 -eq 0 ']'
+ for a in $(seq 100)
+ echo 39
39
+ b=6
+ '[' 6 -eq 0 ']'
+ for a in $(seq 100)
+ echo 40
40
+ b=7
+ '[' 7 -eq 0 ']'
+ for a in $(seq 100)
+ echo 41
41
+ b=8
+ '[' 8 -eq 0 ']'
+ for a in $(seq 100)
+ echo 42
42
```

```
+ b=9
+ '[' 9 -eq 0 ']'
+ for a in $(seq 100)
+ echo 43
43
+ b=10
+ '[' 10 -eq 0 ']'
+ for a in $(seq 100)
+ echo 44
44
+ b=0
+ '[' 0 -eq 0 ']'
+ arr[$a]=44
+ for a in $(seq 100)
+ echo 45
45
+ b=1
+ '[' 1 -eq 0 ']'
+ for a in $(seq 100)
+ echo 46
46
+ b=2
+ '[' 2 -eq 0 ']'
+ for a in $(seq 100)
+ echo 47
47
+ b=3
+ '[' 3 -eq 0 ']'
+ for a in $(seq 100)
+ echo 48
48
+ b=4
+ '[' 4 -eq 0 ']'
+ for a in $(seq 100)
+ echo 49
49
+ b=5
+ '[' 5 -eq 0 ']'
+ for a in $(seq 100)
+ echo 50
50
+ b=6
+ '[' 6 -eq 0 ']'
+ for a in $(seq 100)
+ echo 51
51
+ b=7
+ '[' 7 -eq 0 ']'
+ for a in $(seq 100)
+ echo 52
52
+ b=8
+ '[' 8 -eq 0 ']'
+ for a in $(seq 100)
+ echo 53
53
+ b=9
+ '[' 9 -eq 0 ']'
+ for a in $(seq 100)
+ echo 54
54
+ b=10
+ '[' 10 -eq 0 ']'
+ for a in $(seq 100)
+ echo 55
55
+ b=0
+ '[' 0 -eq 0 ']'
+ arr[$a]=55
+ for a in $(seq 100)
+ echo 56
56
+ b=1
+ '[' 1 -eq 0 ']'
+ for a in $(seq 100)
+ echo 57
57
+ b=2
+ '[' 2 -eq 0 ']'
+ for a in $(seq 100)
+ echo 58
58
+ b=3
+ '[' 3 -eq 0 ']'
+ for a in $(seq 100)
+ echo 59
```

```
59
+ b=4
+ '[' 4 -eq 0 ']'
+ for a in $(seq 100)
+ echo 60
60
+ b=5
+ '[' 5 -eq 0 ']'
+ for a in $(seq 100)
+ echo 61
61
+ b=6
+ '[' 6 -eq 0 ']'
+ for a in $(seq 100)
+ echo 62
62
+ b=7
+ '[' 7 -eq 0 ']'
+ for a in $(seq 100)
+ echo 63
63
+ b=8
+ '[' 8 -eq 0 ']'
+ for a in $(seq 100)
+ echo 64
64
+ b=9
+ '[' 9 -eq 0 ']'
+ for a in $(seq 100)
+ echo 65
65
+ b=10
+ '[' 10 -eq 0 ']'
+ for a in $(seq 100)
+ echo 66
66
+ b=0
+ '[' 0 -eq 0 ']'
+ arr[$a]=66
+ for a in $(seq 100)
+ echo 67
67
+ b=1
+ '[' 1 -eq 0 ']'
+ for a in $(seq 100)
+ echo 68
68
+ b=2
+ '[' 2 -eq 0 ']'
+ for a in $(seq 100)
+ echo 69
69
+ b=3
+ '[' 3 -eq 0 ']'
+ for a in $(seq 100)
+ echo 70
70
+ b=4
+ '[' 4 -eq 0 ']'
+ for a in $(seq 100)
+ echo 71
71
+ b=5
+ '[' 5 -eq 0 ']'
+ for a in $(seq 100)
+ echo 72
72
+ b=6
+ '[' 6 -eq 0 ']'
+ for a in $(seq 100)
+ echo 73
73
+ b=7
+ '[' 7 -eq 0 ']'
+ for a in $(seq 100)
+ echo 74
74
+ b=8
+ '[' 8 -eq 0 ']'
+ for a in $(seq 100)
+ echo 75
75
+ b=9
+ '[' 9 -eq 0 ']'
+ for a in $(seq 100)
+ echo 76
```

```
76
+ b=10
+ '[' 10 -eq 0 ']'
+ for a in $(seq 100)
+ echo 77
77
+ b=0
+ '[' 0 -eq 0 ']'
+ arr[$a]=77
+ for a in $(seq 100)
+ echo 78
78
+ b=1
+ '[' 1 -eq 0 ']'
+ for a in $(seq 100)
+ echo 79
79
+ b=2
+ '[' 2 -eq 0 ']'
+ for a in $(seq 100)
+ echo 80
80
+ b=3
+ '[' 3 -eq 0 ']'
+ for a in $(seq 100)
+ echo 81
81
+ b=4
+ '[' 4 -eq 0 ']'
+ for a in $(seq 100)
+ echo 82
82
+ b=5
+ '[' 5 -eq 0 ']'
+ for a in $(seq 100)
+ echo 83
83
+ b=6
+ '[' 6 -eq 0 ']'
+ for a in $(seq 100)
+ echo 84
84
+ b=7
+ '[' 7 -eq 0 ']'
+ for a in $(seq 100)
+ echo 85
85
+ b=8
+ '[' 8 -eq 0 ']'
+ for a in $(seq 100)
+ echo 86
86
+ b=9
+ '[' 9 -eq 0 ']'
+ for a in $(seq 100)
+ echo 87
87
+ b=10
+ '[' 10 -eq 0 ']'
+ for a in $(seq 100)
+ echo 88
88
+ b=0
+ '[' 0 -eq 0 ']'
+ arr[$a]=88
+ for a in $(seq 100)
+ echo 89
89
+ b=1
+ '[' 1 -eq 0 ']'
+ for a in $(seq 100)
+ echo 90
90
+ b=2
+ '[' 2 -eq 0 ']'
+ for a in $(seq 100)
+ echo 91
91
+ b=3
+ '[' 3 -eq 0 ']'
+ for a in $(seq 100)
+ echo 92
92
+ b=4
+ '[' 4 -eq 0 ']'
+ for a in $(seq 100)
+ echo 93
93
+ b=5
+ '[' 5 -eq 0 ']'
+ for a in $(seq 100)
+ echo 94
94
+ b=6
+ '[' 6 -eq 0 ']'
+ for a in $(seq 100)
+ echo 95
95
+ b=7
+ '[' 7 -eq 0 ']'
+ for a in $(seq 100)
+ echo 96
96
+ b=8
+ '[' 8 -eq 0 ']'
+ for a in $(seq 100)
+ echo 97
97
+ b=9
+ '[' 9 -eq 0 ']'
+ for a in $(seq 100)
+ echo 98
98
+ b=10
+ '[' 10 -eq 0 ']'
+ for a in $(seq 100)
+ echo 99
99
+ b=0
+ '[' 0 -eq 0 ']'
+ arr[$a]=99
+ for a in $(seq 100)
+ echo 100
100
+ b=1
+ '[' 1 -eq 0 ']'
+ echo 11 22 33 44 55 66 77 88 99
11 22 33 44 55 66 77 88 99
```