# Dictionary

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

a. **Roll a dice and find the number between 1-6**
b. **Repeat the dice roll and find the result each time**
c. **Store the result in a dictionary**
d. **Repeat till any one of the number has reached 10 times**
e. **Find the number that reached maximum times and the one that was for minimum times**

**Solution :**

**$ cat Q1.sh**

```
#!/bin/bash
function random() {
        randomNumber=$((1+RANDOM%6))
        echo $randomNumber
}
declare -A dice
a=0
b=0
c=0
d=0
e=0
f=0
ul=10
valid=true
while [ valid ]
do
        case $(random) in
        "1")
                if [ $a -eq $ul ]
                then
                        max=$a
                        break
                else
                        a=$(($a+1))
                        dice[1]=$a
                fi
        ;;
        "2")
                if [ $b -eq $ul ]
                then
```

```
                max=$b
                break
        else
                b=$(($b+1))
        dice[2]=$b
        fi
;;
"3")
        if [ $c -eq $ul ]
        then
                max=$c
                break
        else
                c=$(($c+1))
                dice[3]=$c
        fi
;;
"4")
        if [ $d -eq $ul ]
        then
                max=$d
                break
        else
                d=$(($d+1))
                dice[4]=$d
        fi
;;
"5")
        if [ $e -eq $ul ]
        then
                max=$e
                break
        else
                e=$(($e+1))
                dice[5]=$e
        fi
;;
"6")
        if [ $f -eq $ul ]
        then
                max=$f
                break
        else
```

```
                f=$(($f+1))
                dice[6]=$f
            fi
        ;;
        *)
            echo "Invalid Entry"
        esac
done
echo "Key       Values"
for (( i=1;i<=6;i++ ))
do
        echo "Dice[$i]  ${dice[$i]}"
done
length=${#dice[@]}
minimum=${dice[1]}
for (( i=1; i<=$length; i++ ))
do
        temp=${dice[$i]}
        if [ $temp -eq 10 ]
        then
            echo "Maximum :- $i : ${dice[$i]}"
        fi
done
for (( i=1; i<=$length; i++ ))
do
        temp=${dice[$i]}
        if [ $minimum -gt $temp ]
        then
            minimum=$temp
        fi
done
for (( i=1;i<=$length;i++ ))
do
        temp=${dice[$i]}
        if [ $temp -eq $minimum ]
        then
            echo "Minimum :- $i : ${dice[$i]}"
        fi
done
```

**$ ./Q1.sh**

```
Key     Values
Dice[1] 6
```

```
Dice[2] 10
Dice[3] 4
Dice[4] 6
Dice[5] 5
Dice[6] 9
Maximum :- 2 : 10
Minimum :- 3 : 4
```

**Ques 2. Write a program to generate a birth month of 50 individuals between the year 92 & 93. Find all the individuals having birthdays in the same month. Store it to finally print.**

**Solution :**

**$ cat Q2.sh**

```
#!/bin/bash -x
read -p "Enter the number of person : " num
declare -A month
a=0
b=0
c=0
d=0
e=0
f=0
g=0
h=0
i=0
j=0
k=0
l=0
for (( x=1;x<=$num;x++ ))
do
        randomNumber=$(( RANDOM % 12 + 1 ))
        echo $randomNumber
        case $randomNumber in
        "1")
           a=$(($a+1))
           month[1]=$a
        ;;
        "2")
           b=$(($b+1))
           month[2]=$b
```

```
;;
"3")
   c=$(($c+1))
   month[3]=$c
;;
"4")
   d=$(($d+1))
   month[4]=$d
;;
"5")
   e=$(($e+1))
   month[5]=$e
;;
"6")
   f=$(($f+1))
   month[6]=$f
;;
"7")
   g=$(($g+1))
   month[7]=$g
;;
"8")
   h=$(($h+1))
   month[8]=$h
;;
"9")
   i=$(($i+1))
   month[9]=$i
;;
"10")
   j=$(($j+1))
   month[10]=$j
;;
"11")
   k=$(($k+1))
   month[11]=$k
;;
"12")
   l=$(($l+1))
   month[12]=$l
;;
*)
      echo "Invalid Entry"
```

```
        esac
done
echo "Values  ${!month[@]} |    ${month[@]}"
echo "Month     Count"
for (( i=1;i<=12;i++ ))
do
        echo "Month_[$i]  ${month[$i]}"
done
```

**$ ./Q2.sh**

```
Enter the number of person : 50
10
2
11
1
4
10
11
2
11
12
12
1
3
4
2
1
7
2
6
5
4
5
1
4
8
2
9
6
8
10
8
3
```

9
8
7
5
5
10
9
10
2
9
3
6
5
10
12
2
9
8
Values  12 11 10 1 2 3 4 5 6 7 8 9 |  3 3 6 4 7 3 4 5 3 2 5 5
Month  Count
Month_[1]  4
Month_[2]  7
Month_[3]  3
Month_[4]  4
Month_[5]  5
Month_[6]  3
Month_[7]  2
Month_[8]  5
Month_[9]  5
Month_[10]  6
Month_[11]  3
Month_[12]  3