Dictionary Practice Problems

1) Write a program in the following steps
    a) Roll a die and find the number between 1 to 6
    b) Repeat the Die roll and find the result each time
    c) Store the result in a dictionary
    d) Repeat till one of the numbers has reached 10 times
    e) Find the number that reached maximum times and the one that was for minimum times.

```
#!/bin/bash -x
month=(january february march april may june july august september october november
december )
countOne=0;
countTwo=0;
countThree=0;
countFour=0;
countFive=0;
countSix=0;
countSeven=0;
countEight=0;
countNine=0;
countTen=0;
countEleven=0;
countTwelve=0;

declare -A birthDictionary
for((person=1;person<=50;person++))
do
  birthMonth=$((RANDOM%12+1))
  case $birthMonth in
  1)
    ((countOne++))
    birthDictionary[${month[0]}]=$countOne
    ;;
  2)
    ((countTwo++))
    birthDictionary[${month[1]}]=$countTwo
    ;;
  3)
    ((countThree++))
    birthDictionary[${month[2]}]=$countThree
    ;;
```

```
  4)
    ((countFour++))
    birthDictionary[${month[3]}]=$countFour
    ;;
  5)
    ((countFive++))
    birthDictionary[${month[4]}]=$countFive
    ;;
  6)
    ((countSix++))
    birthDictionary[${month[5]}]=$countSix
    ;;
  7)
    ((countSeven++))
    birthDictionary[${month[6]}]=$countSeven
    ;;
  8)
    ((countEight++))
    birthDictionary[${month[7]}]=$countEight
    ;;
  9)
    ((countNine++))
    birthDictionary[${month[8]}]=$countNine
    ;;
  10)
    ((countTen++))
    birthDictionary[${month[9]}]=$countTen
    ;;
  11)
    ((countEleven++))
    birthDictionary[${month[10]}]=$countEleven
    ;;
  12)
    ((countTwelve++))
    birthDictionary[${month[11]}]=$countTwelve
    ;;
  esac
done
echo "Birth dictionary is:" ${birthDictionary[@]};
```

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

```
#!/bin/bash -x

firstDiceCount=0;
secondDiceCount=0;
thirdDiceCount=0;
fourthDiceCount=0;
fifthDiceCount=0;
sixthDiceCount=0;

  while [ $firstDiceCount -lt 10 -a $secondDiceCount -lt 10 -a $thirdDiceCount -lt 10 -a $fo>
  do
  diceNumber=$(( (RANDOM % 6 ) + 1 ));
  declare -A diceDictionary;

  if [ $diceNumber -eq 1 ]
  then
    (( firstDiceCount++ ));
    diceDictionary[1]="$firstDiceCount";
  elif [ $diceNumber -eq 2 ]
  then
    (( secondDiceCount++ ));
    diceDictionary[2]="$secondDiceCount";
  elif [ $diceNumber -eq 3 ]
  then
    (( thirdDiceCount++ ));
    diceDictionary[3]="$thirdDiceCount";
  elif [ $diceNumber -eq 4 ]
  then
    (( fourthDiceCount++ ));
    diceDictionary[4]="$fourthDiceCount";
  elif [ $diceNumber -eq 5 ]
  then
    (( fifthDiceCount++ ));
    diceDictionary[5]="$fifthDiceCount";
  else
    (( sixthDiceCount++ ));
    diceDictionary[6]="$sixthDiceCount";
  fi
  done
```

```
minDiceNumber=11;
key=0;
for (( index=1; index<=${#diceDictionary[@]}; index++ ))
do
  if [ ${diceDictionary[$index]} -eq 10 ]
  then
    echo "Maximum Dice number $index appears for 10 times";
  fi

  if [ ${diceDictionary[$index]} -lt $minDiceNumber ]
  then
  minDiceNumber=${diceDictionary[$index]};
  key=$index;
  fi
done
 echo "Minimum Dice number $key  appears for $minDiceNumber times";

echo "${diceDictionary[@]}";
echo "${!diceDictionary[@]}";
```