Day 8 Practice Problems

Solved by: SAPTAK DAS(saptakds@gmail.com)

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 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

```
#!/bin/bash -x
declare -A results
for num in `seq 1 6`
do
   results[$num]=0
done
max=0
while [ 0 ]
do
   randomNo=$((1+$RANDOM%6))
   ((results[$randomNo]++))
   if [ ${results[$randomNo]} -eq 10 ]
   then
      max=$randomNo
      break
   fi
done
echo "Number to reach maximum times: $max"
min=1
minOcc=10
for i in `echo ${!results[@]}`
do
   if [ ${results[$i]} -lt $minOcc ]
   then
      minOcc=${results[$i]}
      min=$i
   fi
done
echo "Number to reach minimum times: $min"
```

```
+ declare -A results
++ seq 1 6
+ for num in `seq 1 6`
+ results[$num]=0
+ for num in `seq 1 6`
+ results[$num]=0
+ for num in `seq 1 6`
+ results[$num]=0
+ for num in `seq 1 6`
+ results[$num]=0
+ for num in `seq 1 6`
+ results[$num]=0
+ for num in `seq 1 6`
+ results[$num]=0
+ max=0
+ '[' 0 ']'
+ randomNo=1
+ (( results[1]++ ))
+ '[' 1 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=3
+ (( results[3]++ ))
+ '[' 1 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=2
+ (( results[2]++ ))
+ '[' 1 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=3
+ (( results[3]++ ))
+ '[' 2 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=4
+ (( results[4]++ ))
+ '[' 1 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=5
+ (( results[5]++ ))
+ '[' 1 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=1
+ (( results[1]++ ))
+ '[' 2 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=5
+ (( results[5]++ ))
+ '[' 2 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=5
+ (( results[5]++ ))
+ '[' 3 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=4
+ (( results[4]++ ))
+ '[' 2 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=6
+ (( results[6]++ ))
+ '[' 1 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=2
+ (( results[2]++ ))
+ '[' 2 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=5
+ (( results[5]++ ))
+ '[' 4 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=2
+ (( results[2]++ ))
+ '[' 3 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=1
+ (( results[1]++ ))
+ '[' 3 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=3
+ (( results[3]++ ))
+ '[' 3 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=5
+ (( results[5]++ ))
+ '[' 5 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=1
+ (( results[1]++ ))
+ '[' 4 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=3
+ (( results[3]++ ))
+ '[' 4 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=3
+ (( results[3]++ ))
+ '[' 5 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=6
+ (( results[6]++ ))
+ '[' 2 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=4
+ (( results[4]++ ))
+ '[' 3 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=3
+ (( results[3]++ ))
+ '[' 6 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=1
+ (( results[1]++ ))
+ '[' 5 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=6
+ (( results[6]++ ))
+ '[' 3 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=6
+ (( results[6]++ ))
+ '[' 4 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=2
+ (( results[2]++ ))
+ '[' 4 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=3
+ (( results[3]++ ))
+ '[' 7 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=3
+ (( results[3]++ ))
+ '[' 8 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=5
+ (( results[5]++ ))
+ '[' 6 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=2
+ (( results[2]++ ))
+ '[' 5 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=1
+ (( results[1]++ ))
+ '[' 6 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=6
+ (( results[6]++ ))
+ '[' 5 -eq 10 ']'
+ '[' 0 ']'
```

```
+ randomNo=2
+ (( results[2]++ ))
+ '[' 6 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=3
+ (( results[3]++ ))
+ '[' 9 -eq 10 ']'
+ '[' 0 ']'
+ randomNo=3
+ (( results[3]++ ))
+ '[' 10 -eq 10 ']'
+ max=3
+ break
+ echo 'Number to reach maximum times: 3'
Number to reach maximum times: 3
+ min=1
+ minOcc=10
++ echo 1 2 3 4 5 6
+ for i in `echo ${!results[@]}`
+ '[' 6 -lt 10 ']'
+ minOcc=6
+ min=1
+ for i in `echo ${!results[@]}`
+ '[' 6 -lt 6 ']'
+ for i in `echo ${!results[@]}`
+ '[' 10 -lt 6 ']'
+ for i in `echo ${!results[@]}`
+ '[' 3 -lt 6 ']'
+ minOcc=3
+ min=4
+ for i in `echo ${!results[@]}`
+ '[' 6 -lt 3 ']'
+ for i in `echo ${!results[@]}`
+ '[' 5 -lt 3 ']'
+ echo 'Number to reach minimum times: 4'
Number to reach minimum times: 4
```

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.

```
#!/bin/bash -x
function getMonth(){
    case $1 in
        1)
            echo "January"
        ;;
        2)
            echo "February"
        ;;
        3)
            echo "March"
        ;;
        4)
            echo "April"
        ;;
        5)
            echo "May"
        ;;
        6)
            echo "June"
        ;;
        7)
            echo "July"
        ;;
        8)
            echo "August"
        ;;
        9)
            echo "September"
        ;;
        10)
            echo "October"
        ;;
        11)
            echo "November"
        ;;
        12)
            echo "December"
        ;;
    esac
}
declare -A monthBday
for num in `seq 1 12`
do
    month=(`getMonth $num`)
    monthBday[$month]=""
done
for i in `seq 1 50`
do
```

```
    randomNo=$((1+$RANDOM%12))
    month=(`getMonth $randomNo`)
    monthBday[$month]="${monthBday[$month]} $i"
done
for j in `echo ${!monthBday[@]}`
do
    echo "Person IDs having bdays in $j: ${monthBday[$j]}"
done
```

```
Person IDs having bdays in January:  6 10 12 15 49
Person IDs having bdays in December:  19
Person IDs having bdays in November:  33 39 41
Person IDs having bdays in May:  1 4 13 24 28 29 30 36 42
Person IDs having bdays in April:  2 7 8 14 22 32 45
Person IDs having bdays in September:  35 48
Person IDs having bdays in February:  18 26 31
Person IDs having bdays in March:  20 21 34 40 50
Person IDs having bdays in June:  17 23 27 37
Person IDs having bdays in August:  3 16 43 46
Person IDs having bdays in October:  25 38 47
Person IDs having bdays in July:  5 9 11 44
```