

day-5 Practiceproblem

Answer Sheet



June 27, 2021

Akash suchal

Suchak.akash@gmail.com

SEQUENCES PRACTICE PROBLEMS

1 – USE RANDOM FUCTION (( RANDOM )) TO GET SINGLE DIGIT

ANS :

#!/bin/bash -x

ranCheck=$((RANDOM%10))

echo "$ranCheck"

OUTPUT :

$ ./1-ranFunction.sh

+ ranCheck=0

+ echo 0

0

2 – USE RANDOM TO GET DICE NUMBER BETWEEN 1 TO 6

ANS :

#!/bin/bash -x

diceNumber=$(((RANDOM%6)+1))

echo "$diceNumber"

OUTPUT :

$ ./2-diceNumber.sh

+ diceNumber=2

+ echo 2

2

3 – ADD TWO RANDOM DICE NUMBER AND PRINT THE RESULT

ANS :

#!/bin/bash

#Assign Value for 2 Dices

diceNumber1=$(((RANDOM%6)+1))

diceNumber2=$(((RANDOM%6)+1))

#Add Two Dice Number

result=$(($diceNumber1+$diceNumber2))

#Display Result

echo "1st Dice Number is : $diceNumber1"

echo "2nd Dice Number is : $diceNumber2"

echo "Total : $result"

OUTPUT :

$ ./3-twoDiceNumber.sh

1st Dice Number is : 2

2nd Dice Number is : 5

Total : 7

4 – WRITE A PROGRAM THAT READS 5 RANDOM 2 DIGIT VALUES, THEN FIND THEIR SUM AND AVERAGE

ANS :

#!/bin/bash

#Write a program that reads 5 Random 2 Digit values, then find

#their sum and average

#Assign value for Iteration

iterationValue=5

#Generate RandomValues and Make Sum of Random Value

for (( i=1; i<=$iterationValue; i++ ))

do

ranNumber=$(((RANDOM%90)+10))

echo "Random Value $i : $ranNumber"

sum=$(($sum+$ranNumber))

done

#Get average

avg=`awk -v sum=$sum -v iv=$iterationValue 'BEGIN { print (sum/iv) }'`

#avg=$(($sum/$iterationValue))

#Display Output

echo "Sum of 5 Random Values : $sum"

echo "Average : $avg"

OUTPUT :

$ ./4-twoDigitValueOperations.sh

Random Value 1 : 48

Random Value 2 : 81

Random Value 3 : 44

Random Value 4 : 97

Random Value 5 : 27

Sum of 5 Random Values : 297

Average : 59.4

5 – UNIT CONVERSION

1. 1 ft = 12 in then 42 in = ? ft
2. Rectangular plot of 60 feet \* 40 feet in meters
3. Calculate area of 25 such plots In acres

ANS :

#!/bin/bash

#Unit Conversion

#a. 1ft = 12 in then 42 in = ?ft

result=`awk -v fixFt=1 -v fixIn=12 -v checkFt=42 'BEGIN {print (checkFt\*fixFt)/fixIn }'`

echo "1ft = 12 in"

echo "Ans : 42 in = $result ft"

echo "----------------------------------------------------------------------"

#b. Rectangular plot of 60 feet \* 40 feet in meters

echo "Length = 60 feet \* Width = 40 feet in Meter "

areaMeter=`awk -v len=60 -v wid=40 'BEGIN {print ( len \*wid ) / 3.28084}'`

#echo "Area in feet : $areaFeet feet"

#areaMeter=$(($areaFeet/3.28084))

echo "In Meter : $areaMeter"

echo "----------------------------------------------------------------------"

#C. calculate area of 25 such plots in acres

totalAreaFeet=$((60\*40\*25))

echo "Area of 25 plots (in feet) : $totalAreaFeet"

totalAreaAcre=`awk -v totalAreaFeet=$totalAreaFeet 'BEGIN {print (totalAreaFeet / 43560)}'`

echo "Area of 25 plots (in Acre) : $totalAreaAcre"

OUTPUT :

$ ./5-unitConversion.sh

1ft = 12 in

Ans : 42 in = 3.5 ft

----------------------------------------------------------------------

Length = 60 feet \* Width = 40 feet in Meter

In Meter : 731.52

----------------------------------------------------------------------

Area of 25 plots (in feet) : 60000

Area of 25 plots (in Acre) : 1.37741

SELECTION PRACTICE PROBLEMS

1 – WRITE A PROGRAM THAT READS 5 RANDOM DIGITS VALUES AND THEN OUTPUTS THE MINIMUM AND THE MAXIMUM VALUE

ANS :

#!/bin/bash

iterationValue=5

min\_value=0

max\_value=0

for (( i=1; i<=$iterationValue; i++ ))

do

randomCheck=$(((RANDOM%900)+100))

echo "Random Value $i : $randomCheck"

if [ $max\_value -lt $randomCheck ]

then

max\_value=$randomCheck

fi

if [ $min\_value -eq 0 ]

then

min\_value=$randomCheck

elif [ $min\_value -gt $randomCheck ]

then

min\_value=$randomCheck

fi

done

echo "Minimum Value : $min\_value"

echo "Maximum Value : $max\_value"

OUTPUT :

$ ./1.Random3Digit.sh

Random Value 1 : 727

Random Value 2 : 656

Random Value 3 : 918

Random Value 4 : 232

Random Value 5 : 185

Minimum Value : 185

Maximum Value : 918

2 – WRITE A PROGRAM THAT TAKES DAY AND MONTH FROM THE COMMAND LINE AND PRINTS TRUE IF DAY OF MONTH IS BETWEEN MARCH 20 AND JUNE 20, FALSE OTHERWISE

ANS :

#!/bin/bash -x

read -p "Enter Month (eg. january ): " month

read -p "Enter Date (eg. 21 ) : " date

if [ $month == 'march' ] && [ $date -ge 20 ] && [ $date -le 31 ]

then

echo " True "

elif [ $month == 'april' ] && [ $date -ge 1 ] && [ $date -le 30 ]

then

echo " True "

elif [ $month == 'may' ] && [ $date -ge 1 ] && [ $date -le 31 ]

then

echo " True "

elif [ $month == 'june' ] && $date -ge 1 ] && [ $date -le 20 ]

then

echo " True "

else

echo " Invlid date "

fi

OUTPUT :

$ ./2.DayAndMonth.sh

+ read -p 'Enter Month (eg. january ): ' month

Enter Month (eg. january ): march

+ read -p 'Enter Date (eg. 21 ) : ' date

Enter Date (eg. 21 ) : 30

+ '[' march == march ']'

+ '[' 30 -ge 20 ']'

+ '[' 30 -le 31 ']'

+ echo ' True '

True

3 – WRITE A PROGRAM THAT TAKES A YEAR AS INPUT AND OUTPUTS THE YEAR IS A LEAP YEAR OR NOT A LEAP YEAR. A LEAP YEAR CHECKS FOR 4 DIGITS NUMBER, DIVISIBLE BY 4 AND NOT 100 UNLESS DIVISIBLE BY 400.

ANS :

#!/bin/bash -x

echo "Leap year or Not"

read -p "Enter 4 Digit Year : " year

count=${#year}

if [ $count == 4 ]

then

if [ $(($year%4)) -eq 0 ] && [ $(($year%100)) -ne 0 -o $(($year%400)) -eq 0 ]

then

echo "$year is a Leap Year"

else

echo "$year is Not Leap Year"

fi

else

echo "Invalid Input !!!"

fi

OUTPUT :

$ ./3.LeapYearOrNot.sh

+ echo 'Leap year or Not'

Leap year or Not

+ read -p 'Enter 4 Digit Year : ' year

Enter 4 Digit Year : 2010

+ count=4

+ '[' 4 == 4 ']'

+ '[' 2 -eq 0 ']'

+ echo '2010 is Not Leap Year'

2010 is Not Leap Year

4 – WRITE A PROGRAM TO SIMULATE A COIN FLIP AND PRINT OUT “HEADS” OR “TAILS” ACCORDINGLY

ANS :

#!/bin/bash -x

flip=$((RANDOM%2))

if [ $flip -eq 0 ]

then

echo "HeadS"

else

echo "TailS"

fi

OUTPUT :

$ ./4.coinFlip.sh

+ flip=1

+ '[' 1 -eq 0 ']'

+ echo TailS

TailS

SELECTION PRACTICE PROBLEMS WITH if, elif and else

1 – READ A SINGLE DIGIT NUMBER AND WRITE THE NUMBER IN WORD

ANS :

#!/bin/bash -x

read -p "Enter Single Digit Number : " num

checkNum=${#num}

if [ $checkNum -eq 1 ]; then

if [ $num -eq 0 ]; then

echo "Zero";

elif [ $num -eq 1 ]; then

echo "ONE";

elif [ $num -eq 2 ]; then

echo "TWO"

elif [ $num -eq 3 ]; then

echo "THREE"

elif [ $num -eq 4 ]; then

echo "FOUR"

elif [ $num -eq 5 ]; then

echo "FIVE"

elif [ $num -eq 6 ]; then

echo "SIX"

elif [ $num -eq 7 ]; then

echo "SEVEN"

elif [ $num -eq 8 ]; then

echo "EIGHT"

elif [ $num -eq 9 ]; then

echo "NINE"

fi

else

echo "Invalid Input !!!"

fi

OUTPUT :

$ ./1-SingleDigitNum.sh

+ read -p 'Enter Single Digit Number : ' num

Enter Single Digit Number : 5

+ checkNum=1

+ '[' 1 -eq 1 ']'

+ '[' 5 -eq 0 ']'

+ '[' 5 -eq 1 ']'

+ '[' 5 -eq 2 ']'

+ '[' 5 -eq 3 ']'

+ '[' 5 -eq 4 ']'

+ '[' 5 -eq 5 ']'

+ echo FIVE

FIVE

2 – READ A NUMBER AND DISPLAY THE WEEK DAY (SUNDAY, MONDAY, …)

ANS :

#!/bin/bash -x

read -p "Enter Number between 1 to 7 : " num

checkNum=${#num}

if [ $checkNum -eq 1 ]; then

if [ $num -eq 1 ]; then

echo "Sunday"

elif [ $num -eq 2 ]; then

echo "Monday"

elif [ $num -eq 3 ]; then

echo "Tuesday"

elif [ $num -eq 4 ]; then

echo "Wednesday"

elif [ $num -eq 5 ]; then

echo "Thursday"

elif [ $num -eq 6 ]; then

echo "Friday"

elif [ $num -eq 7 ]; then

echo "Saturday"

fi

else

echo "Invalid Input !!!"

fi

OUTPUT :

$ ./2-WeekDays.sh

+ read -p 'Enter Number between 1 to 7 : ' num

Enter Number between 1 to 7 : 6

+ checkNum=1

+ '[' 1 -eq 1 ']'

+ '[' 6 -eq 1 ']'

+ '[' 6 -eq 2 ']'

+ '[' 6 -eq 3 ']'

+ '[' 6 -eq 4 ']'

+ '[' 6 -eq 5 ']'

+ '[' 6 -eq 6 ']'

+ echo Friday

Friday

3 – READ A NUMBER 1,10,100,1000 ETC AND DISPLAY UNIT, TEN, HUNDRED, …

ANS :

#!/bin/bash -x

read -p "Enter Number (1,10,100, ...) : " num

if [ $num -eq 1 ]; then

echo "One"

elif [ $num -eq 10 ]; then

echo "Ten"

elif [ $num -eq 100 ]; then

echo "Hundred"

elif [ $num -eq 1000 ]; then

echo "Thousand"

elif [ $num -eq 10000 ]; then

echo "Ten Thousand"

elif [ $num -eq 100000 ]; then

echo "One Lakh"

elif [ $num -eq 1000000 ]; then

echo "Ten Lakhs"

elif [ $num -eq 10000000 ]; then

echo "One Crore"

elif [ $num -eq 100000000 ]; then

echo "Ten Crore"

else

echo "Please Enter Valid Number"

fi

OUTPUT :

$ ./3-Number-1-10-100.sh

+ read -p 'Enter Number (1,10,100, ...) : ' num

Enter Number (1,10,100, ...) : 1000000

+ '[' 1000000 -eq 1 ']'

+ '[' 1000000 -eq 10 ']'

+ '[' 1000000 -eq 100 ']'

+ '[' 1000000 -eq 1000 ']'

+ '[' 1000000 -eq 10000 ']'

+ '[' 1000000 -eq 100000 ']'

+ '[' 1000000 -eq 1000000 ']'

+ echo 'Ten Lakhs'

Ten Lakhs

4 – ENTER 3 NUMBERS DO FOLLOWING ARITHMETIC OPERATION AND FIND THE ONE THAT IS MAXIMUM AND MINIMUM

1. a + b \* c
2. a % b + c
3. a + b / c
4. a \* b + c

ANS :

#!/bin/bash -x

read -p "Enter Value of A : " a

read -p "Enter Value of B : " b

read -p "Enter Value of C : " c

num1=$((a+b\*c))

num2=$((a%b+c))

num3=$((a+b/c))

num4=$((a\*b+c))

max=0

min=0

if [ $num1 -ge $num2 ]; then

max=$num1

else

max=$num2

fi

if [ $num3 -ge $max ]; then

max=$num3

elif [ $num4 -ge $max ]; then

max=$num4

fi

if [ $num1 -lt $num2 ]; then

min=$num1

else

min=$num2

fi

if [ $num3 -lt $min ]; then

min=$num3

elif [ $num4 -lt $min ]; then

min=$num4

fi

echo "Maximum Value : " $max

echo "Minimum Value : " $min

OUTPUT :

$ ./4-arithmeticOperation.sh

+ read -p 'Enter Value of A : ' a

Enter Value of A : 10

+ read -p 'Enter Value of B : ' b

Enter Value of B : 20

+ read -p 'Enter Value of C : ' c

Enter Value of C : 30

+ num1=610

+ num2=40

+ num3=10

+ num4=230

+ max=0

+ min=0

+ '[' 610 -ge 40 ']'

+ max=610

+ '[' 10 -ge 610 ']'

+ '[' 230 -ge 610 ']'

+ '[' 610 -lt 40 ']'

+ min=40

+ '[' 10 -lt 40 ']'

+ min=10

+ echo 'Maximum Value : ' 610

Maximum Value : 610

+ echo 'Minimum Value : ' 10

Minimum Value : 10

SELECTION PRACTICE PROBLEMS WITH CASE STATEMENT

1 – READ A SINGLE DIGIT NUMBER AND WRITE THE NUMBER IN WORD USING CASE

ANS :

#!/bin/bash -x

read -p "Enter Single Digit Number : " num

checkNum=${#num}

if [ $checkNum -eq 1 ]; then

case $num in

1) echo "ONE"

;;

2) echo "TWO"

;;

3) echo "THREE"

;;

4) echo "FOUR"

;;

5) echo "FIVE"

;;

6) echo "SIX"

;;

7) echo "SEVEN"

;;

8) echo "EIGHT"

;;

9) echo "NINE"

;;

\*) echo "Enter Valid Input !!"

;;

esac

else

echo "Invalid Input"

fi

OUTPUT :

$ ./1-case-SingleDigitNum.sh

+ read -p 'Enter Single Digit Number : ' num

Enter Single Digit Number : 6

+ checkNum=1

+ '[' 1 -eq 1 ']'

+ case $num in

+ echo SIX

SIX

2 – READ A NUMBER AND DISPLAY THE WEEK DAY (SUNDAY, MONDAY …)

ANS :

#!/bin/bash -x

read -p "Enter Number Between 1 to 7 : " num

case $num in

1) echo "Sunday"

;;

2) echo "Monday"

;;

3) echo "Tuesday"

;;

4) echo "Wednesday"

;;

5) echo "Thursday"

;;

6) echo "Friday"

;;

7) echo "Saturday"

;;

\*) echo "Invalid Input !!!"

;;

esac

OUTPUT :

$ ./2-case-WeekDays.sh

+ read -p 'Enter Number Between 1 to 7 : ' num

Enter Number Between 1 to 7 : 6

+ case $num in

+ echo Friday

Friday

3 – READ A NUMBER 1, 10, 100, 1000 ….ETC AND DISPLAY UNIT, TEN, HUNDRED,….

ANS :

#!/bin/bash -x

read -p "Enter Number like (1,10,100 ...) : " num

case $num in

1) echo "One"

;;

10) echo "Ten"

;;

100) echo "Hundred"

;;

1000) echo "Thousand"

;;

10000) echo "Ten Thousand"

;;

100000) echo "One Lakh"

;;

1000000) echo "Ten Lakhs"

;;

\*) echo "Out Of Dictionary"

;;

esac

OUTPUT :

$ ./3-case-Number-1-10-100.sh

+ read -p 'Enter Number like (1,10,100 ...) : ' num

Enter Number like (1,10,100 ...) : 1000000

+ case $num in

+ echo 'Ten Lakhs'

Ten Lakhs

4 – WRITE A PROGRAM THAT TAKES USER INPUTS AND DOES UNIT CONVERSION OF DIFFERENT LENGTH UNITS :

1. FEET TO INCH
2. FEET TO METER
3. INCH TO FEET
4. METER TO FEET

ANS :

#!/bin/bash

echo "Press 1 : Feet to Inch"

echo "Press 2 : Feet to Meter"

echo "Press 3 : Inch to Feet"

echo "Press 4 : Meter to Feet"

read -p "Enter Number : " num

checkNum=${#num}

if [ $checkNum -eq 1 ]; then

case $num in

1) read -p "Enter Feet : " feet

inches=12

totalInches=$((feet\*inches))

echo "Feet to Inches : $feet = $totalInches"

;;

2) read -p "Enter Feet : " feet

meter=0.3048

totalMeter=`awk -v feet=$feet -v meter=$meter 'BEGIN {totalMeter=(feet\*meter); print totalMeter}'`

echo "Feet to Meter : $feet = $totalMeter"

;;

3) read -p "Enter Inches : " inches

totalFeet=`awk -v inches=$inches 'BEGIN {totalFeet=(inches/12); print totalFeet }'`

echo "Inches to Feet : $inches = $totalFeet"

;;

4) read -p "Enter Meter : " meter

feet=3.28084

totalFeet=`awk -v feet=$feet -v meter=$meter 'BEGIN {totalFeet=(feet\*meter); print totalFeet}'`

echo "Meter to Feet : $meter = $totalFeet"

;;

\*) echo "Invalid Input !!!"

;;

esac

else

echo "Invalid Input !!!"

fi

OUTPUT :

$ ./4-UnitConversion.sh

Press 1 : Feet to Inch

Press 2 : Feet to Meter

Press 3 : Inch to Feet

Press 4 : Meter to Feet

Enter Number : 1

Enter Feet : 5

Feet to Inches : 5 = 60