**Sequences Practice Problems**

1. Use Random Function (( RANDOM )) to get Single Digit

Solution::-

#!/bin/bash -x

val=$((RANDOM % 10));

echo $val

output;

$ ./SingleDig.sh

+ val=2

+ echo 2

2

#########################################################################

2. Use Random to get Dice Number between 1 to 6

Solution::-

#! /bin/bash -x

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

echo $dice

output:

$ ./Dice.sh

+ dice=1

+ echo 1

1

#####################################################################

3. Add two Random Dice Number and Print the Result

#!/bin/bash -x

dice1=$((RANDOM % 6 + 1));

dice2=$((RANDOM % 6 + 1));

echo "value of fisrt attempt:: "$dice1

echo "value of second attempt:: "$dice2

sum=$(($dice1+$dice2));

echo "addition of two random dice number:: "$sum

OUTPUT:

$ ./TwoDice.sh

+ dice1=2

+ dice2=2

+ echo 'value of fisrt attempt:: 2'

value of fisrt attempt:: 2

+ echo 'value of second attempt:: 2'

value of second attempt:: 2

+ sum=4

+ echo 'addition of two random dice number:: 4'

addition of two random dice number:: 4

#####################################################################

4. Write a program that reads 5 Random 2 Digit values , then find their

sum and the average

solution::

#!/bin/bash -x

sum=0;

for i in {1..5}

do

val1=$(( (RANDOM%89)+10 ));

echo $i times $val1;

sum=$(($val1+$sum));

done

echo "sum of 5 random values :" $sum;

avg=$(($sum/5));

echo "average of 5 random values is" $avg;

output::

$ ./RandomAdd.sh

+ sum=0

+ for i in {1..5}

+ val1=80

+ echo 1 times 80

1 times 80

+ sum=80

+ for i in {1..5}

+ val1=63

+ echo 2 times 63

2 times 63

+ sum=143

+ for i in {1..5}

+ val1=54

+ echo 3 times 54

3 times 54

+ sum=197

+ for i in {1..5}

+ val1=72

+ echo 4 times 72

4 times 72

+ sum=269

+ for i in {1..5}

+ val1=76

+ echo 5 times 76

5 times 76

+ sum=345

+ echo 'sum of 5 random values :' 345

sum of 5 random values : 345

+ avg=69

+ echo 'average of 5 random values is' 69

average of 5 random values is 69

###################################################################

5. Unit Conversion

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

b. Rectangular Plot of 60 feet x 40 feet in meters

c. Calculate area of 25 such plots in acres

Solution :

#!/bin/bash -x

# ft=1/12inches

echo "1 ft = 12inch then 42inch = " $(( 42 / 12 ))

echo "area of rectangular plot 60 feet \* 40 feet in meter "

area=$(( 60\*40 ))

echo $area

areaMeter=$((($area) / 3 ))

echo $areaMeter

echo "area of such 24 plot in meter "

areaMeter=$((($areaMeter)\* 24 ))

acre=$((($areaMeter)/4047))

echo "area in acre of 24 plot" $acre

output::

$ ./qu5\_unit\_conv.sh

+ echo '1 ft = 12inch then 42inch = ' 3

1 ft = 12inch then 42inch = 3

+ echo 'area of rectangular plot 60 feet \* 40 feet in meter '

area of rectangular plot 60 feet \* 40 feet in meter

+ area=2400

+ echo 2400

2400

+ areaMeter=800

+ echo 800

800

+ echo 'area of such 24 plot in meter '

area of such 24 plot in meter

+ areaMeter=19200

+ acre=4

+ echo 'area in acre of 24 plot' 4

area in acre of 24 plot 4

**Selection Practice Problems with if & else**

1. Write a program that reads 5 Random 3 Digit values and then outputs the minimum

and the maximum value

solution::

#!/bin/bash -x

# Creating five Random variables

val1=$((( RANDOM % 899 ) + 100 ))

echo $val1

val2=$((( RANDOM % 899 ) + 100 ))

echo $val2

val3=$((( RANDOM % 899 ) + 100 ))

echo $val3

val4=$((( RANDOM % 899 ) + 100 ))

echo $val4

val5=$((( RANDOM % 899 ) + 100 ))

echo "maximum value.........."

if [[ $val1 -gt $val2 && $val1 -gt $val3 && $val1 -gt $val4 && $val1 -gt $val5 ]]

then

echo " value 1 " $val1 "is maximum value ";

elif [[ $val2 -gt $val1 && $val2 -gt $val3 && $val2 -gt $val4 && $val2 -gt $val5 ]]

then

echo " value 2 " $val2 "is maximum value ";

elif [[ $val3 -gt $val1 && $val3 -gt $val2 && $val3 -gt $val4 && $val3 -gt $val5 ]]

then

echo " value 3 " $val3 "is maximum value ";

elif [[ $val4 -gt $val1 && $val4 -gt $val2 && $val4 -gt $val3 && $val4 -gt $val5 ]]

then

echo " value 4 " $val4 "is maximum value ";

elif [[ $val5 -gt $val1 && $val5 -gt $val2 && $val5 -gt $val3 && $val5 -gt $val4 ]]

then

echo " value 5 " $val5 "is maximum value ";

else

echo "................";

fi

output::

$ ./maxORmin.sh

+ val1=963

+ echo 963

963

+ val2=927

+ echo 927

927

+ val3=764

+ echo 764

764

+ val4=470

+ echo 470

470

+ val5=389

+ echo 'maximum value..........'

maximum value..........

+ [[ 963 -gt 927 ]]

+ [[ 963 -gt 764 ]]

+ [[ 963 -gt 470 ]]

+ [[ 963 -gt 389 ]]

+ echo ' value 1 ' 963 'is maximum value '

value 1 963 is maximum value

+ echo 'minimum value..........'

minimum value..........

+ [[ 963 -lt 927 ]]

+ [[ 927 -lt 963 ]]

+ [[ 927 -lt 764 ]]

+ [[ 764 -lt 963 ]]

+ [[ 764 -lt 927 ]]

+ [[ 764 -lt 470 ]]

+ [[ 470 -lt 963 ]]

+ [[ 470 -lt 927 ]]

+ [[ 470 -lt 764 ]]

+ [[ 470 -lt 389 ]]

+ [[ 389 -lt 963 ]]

+ [[ 389 -lt 927 ]]

+ [[ 389 -lt 764 ]]

+ [[ 389 -lt 470 ]]

+ echo ' value 5 ' 389 'is minimum value '

value 5 389 is minimum value

#####################################################################

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.

Solution:

#!/bin/bash -x

read -p "enter month [1-12]" month

read -p "enter date [ 1-31 ]" day

if [ $month -ge 3 ] && [ $day -ge 20 ]

then

echo "true"

elif [ $month -le 6 ] && [ $day -le 20 ]

then

echo "true"

else

echo "false"

fi

output::

$ ./dayaDndDate1.sh

+ read -p 'enter month [1-12]' month

enter month [1-12]5

+ read -p 'enter date [ 1-31 ]' day

enter date [ 1-31 ]22

+ '[' 5 -ge 3 ']'

+ '[' 22 -ge 20 ']'

+ 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 Digit Number, Divisible by 4 and not 100 unless

divisible by 400.

Solution::

#!/bin/bash -x

echo "check for leap year.."

read -p "enter any YEAR to check year is leap year or not :: " year

echo $year

if [ $(( $year % 400 )) -eq 0 ]

then

echo $year " is a leap year "

elif [ $(( $year % 100 )) -eq 0 ]

then

echo $year "is not a leap year "

elif [ $(( $year % 4 )) -eq 0 ]

then

echo $year " is a leap year "

else

echo $year "is not a leap year "

fi

output::

$ ./leapyear.sh

+ echo 'check for leap year..'

check for leap year..

+ read -p 'enter any YEAR to check year is leap year or not :: ' year

enter any YEAR to check year is leap year or not :: 2016

+ echo 2016

2016

+ '[' 16 -eq 0 ']'

+ '[' 16 -eq 0 ']'

+ '[' 0 -eq 0 ']'

+ echo 2016 ' is a leap year '

2016 is a leap year

#########################################################################

4. Write a program to simulate a coin flip and print out "Heads" or "Tails" accordingly.

Solution ::

#!/bin/bash -x

#assign 0 to the head...

heads=0;

#assign 1 to the tail...

tails=1;

# total count is tail+head(1+1)

totalcount=2;

echo "Flipping the coin"

Flip=$((RANDOM%$totalcount))

echo $Flip

if [ $heads -eq $Flip ]

then

echo "Heads"

else

echo "Tails"

fi

output::

$ ./coin.sh

+ heads=0

+ tails=1

+ totalcount=2

+ echo 'Flipping the coin'

Flipping the coin

+ Flip=1

+ echo 1

1

+ '[' 0 -eq 1 ']'

+ echo Tails

Tails

**Selection Practice Problems with if, elif and else**

1. Read a single digit number and write the number in word

Solution::

#!/bin/bash

read -p "enter any number in range [0-9] :: " digit

if [ $digit -eq 1 ]

then

echo "one"

elif [ $digit -eq 2 ]

then

echo "two "

elif [ $digit -eq 3 ]

then

echo "three "

elif [ $digit -eq 4 ]

then

echo "four "

elif [ $digit -eq 5 ]

then

echo "five "

elif [ $digit -eq 6 ]

then

echo "six "

elif [ $digit -eq 7 ]

then

echo "seven "

elif [ $digit -eq 8 ]

then

echo "eight "

elif [ $digit -eq 9 ]

then

echo "nine "

elif [ $digit -eq 0 ]

then

echo "zero"

output::

$ ./numinword.sh

+ read -p 'enter any number in range [0-9] :: ' digit

enter any number in range [0-9] :: 5

+ '[' 5 -eq 1 ']'

+ '[' 5 -eq 2 ']'

+ '[' 5 -eq 3 ']'

+ '[' 5 -eq 4 ']'

+ '[' 5 -eq 5 ']'

+ echo 'five '

five

1. Read a Number and Display the week day (Sunday, Monday,...)

Solution::

#!/bin/bash -x

read -p "enter week day in number[1 - 7]" weekday

if [ $weekday -eq 1 ]

then

echo "today is SUNDAY "

elif [ $weekday -eq 2 ]

then

echo "today is MONDAY "

elif [ $weekday -eq 3 ]

then

echo "today is TUESDAY"

elif [ $weekday -eq 4 ]

then

echo "today is WEDNESDAY"

elif [ $weekday -eq 5 ]

then

echo "today is THURSDAY"

elif [ $weekday -eq 6 ]

then

echo "today is FRIDAY"

elif [ $weekday -eq 7 ]

then

echo "todat is SATURDAY"

else

echo "not in day [wrong input]"

fi

output::

$ ./displayweek.sh

+ read -p 'enter week day in number[1 - 7]' weekday

enter week day in number[1 - 7]6

+ '[' 6 -eq 1 ']'

+ '[' 6 -eq 2 ']'

+ '[' 6 -eq 3 ']'

+ '[' 6 -eq 4 ']'

+ '[' 6 -eq 5 ']'

+ '[' 6 -eq 6 ']'

+ echo 'today is FRIDAY'

today is FRIDAY

1. Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,...

SOLUTION::

#!/bin/bash -x

read -p "enter any number like [1,10,100,1000]" number

if [ $number -eq 1 ]

then

echo "UNIT"

elif [ $number -eq 10 ]

then

echo "TEN"

elif [ $number -eq 100 ]

then

echo "HUNDRED"

elif [ $number -eq 1000 ]

then

echo "THOUSAND"

elif [ $number -eq 10000 ]

then

echo "TEN THOUSAND"

elif [ $number -eq 100000]

then

echo "LACK"

else

echo "OTHTER"

fi

output::

$ ./unitTest1.sh

+ read -p 'enter any number like [1,10,100,1000]' number

enter any number like [1,10,100,1000]100

+ '[' 100 -eq 1 ']'

+ '[' 100 -eq 10 ']'

+ '[' 100 -eq 100 ']'

+ echo HUNDRED

HUNDRED

######################################################################

4. Enter 3 Numbers do following arithmetic operation and find the one that

is maximum and minimum

1. a + b \* c 3. c + a / b

2. a % b + c 4. a \* b + c

Solution ::

#!/bin/bash -x

read -p "enter value of a :: " a

read -p "enter value of b :: " b

read -p "enter value of c :: " c

exp1=$((a+b\*c))

exp2=$((a%b+c))

exp3=$((c+a/b))

exp4=$((a\*b+c))

echo "expression (a+b\*c)= " $exp1

echo "expression (a%b+c)= " $exp2

echo "expression (c+a/b)= " $exp3

echo "expression (a\*b+c)= " $exp4

# Loop for checking max

if [[ $exp1 -gt $exp2 && $exp1 -gt $exp3 && $exp1 -gt $exp4 ]]

then

echo "exp1 is maximum"

elif [[ $exp2 -gt $exp1 && $exp2 -gt $exp3 && $exp2 -gt $exp4 ]]

then

echo "exp2 is maximum"

elif [[ $exp3 -gt $exp1 && $exp3 -gt $exp2 && $exp3 -gt $exp4 ]]

then

echo "exp3 is maximum"

else

echo "exp4 is max"

fi

# loop ends

# loop for checking min

if [[ $exp1 -lt $exp2 && $exp1 -lt $exp3 && $exp1 -lt $exp4 ]]

then

echo "exp1 is minimum"

elif [[ $exp2 -lt $exp1 && $exp2 -lt $exp3 && $exp2 -lt $exp4 ]]

then

echo "exp2 is minimum"

elif [[ $exp3 -lt $exp1 && $exp3 -lt $exp2 && $exp3 -lt $exp4 ]]

then

echo "exp3 is minimum"

else

echo "exp4 is minimum"

fi

#lopp ends

Output::

$ ./maxorminexp.sh

+ read -p 'enter value of a :: ' a

enter value of a :: 55

+ read -p 'enter value of b :: ' b

enter value of b :: 2

+ read -p 'enter value of c :: ' c

enter value of c :: 33

+ exp1=121

+ exp2=34

+ exp3=60

+ exp4=143

+ echo 'expression (a+b\*c)= ' 121

expression (a+b\*c)= 121

+ echo 'expression (a%b+c)= ' 34

expression (a%b+c)= 34

+ echo 'expression (c+a/b)= ' 60

expression (c+a/b)= 60

+ echo 'expression (a\*b+c)= ' 143

expression (a\*b+c)= 143

+ [[ 121 -gt 34 ]]

+ [[ 121 -gt 60 ]]

+ [[ 121 -gt 143 ]]

+ [[ 34 -gt 121 ]]

+ [[ 60 -gt 121 ]]

+ echo 'exp4 is max'

exp4 is max

+ [[ 121 -lt 34 ]]

+ [[ 34 -lt 121 ]]

+ [[ 34 -lt 60 ]]

+ [[ 34 -lt 143 ]]

+ echo 'exp2 is minimum'

exp2 is minimum

**Selection Practice Problems with case statement**

1. Read a single digit number and write the number in word using Case

Solution::

#!/bin/bash -x

read -p "enter any number between 0-9:: " number

case "$number" in

1)

echo "$number : one";;

2)

echo "$number : two";;

3)

echo "$number : three";;

4)

echo "$number : four";;

5)

echo "$number : five";;

6)

echo "$number : six";;

7)

echo "$number : seven";;

8)

echo "$number : eight";;

9)

echo "$number : nine";;

0)

echo "$number : zero";;

\*)

echo "wrong input ";;

esac

output::

$ ./numinwordCASE.sh

+ read -p 'enter any number between 0-9:: ' number

enter any number between 0-9:: 9

+ case "$number" in

+ echo '9 : nine'

9 : nine

#################################################################

2. Read a Number and Display the week day (Sunday, Monday,...)

Solution::

#!/bin/bash

read -p "enter any number between 1-7 :: " weekday

case "$weekday" in

1)

echo "$weekday : SUNDAY";;

2)

echo "$weekday : MONDAY";;

3)

echo "$weekday : TUESDAY";;

4)

echo "$weekday : WEDNESDAY";;

5)

echo "$weekday : THURSDAY";;

6)

echo "$weekday : FRIDAY";;

7)

echo "$weekday : SATURDAY";;

\*)

echo "$weekday : WRONG INPUT ";;

esac

output::

$ ./displayweekCASE.sh

enter any number between 1-7 :: 1

1 : SUNDAY

###############################################################

1. Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,...

Solution::

#!/bin/bash -x

read -p "enter any number like [1,10,100,1000]" number

case $number in

1)

echo "Unit" ;;

10)

echo "Ten" ;;

100)

echo "Hundred" ;;

1000)

echo "Thousand " ;;

10000)

echo "Ten Thousand " ;;

100000)

echo "Lack " ;;

\*)

echo "Other ";;

esac

output::

$ ./unitTestCase.sh

+ read -p 'enter any number like [1,10,100,1000]' number

enter any number like [1,10,100,1000]1000

+ case $number in

+ echo 'Thousand '

Thousand

##################################################################

4. Write a program that takes User Inputs and does Unit Conversion of

different Length units

1. Feet to Inch 3. Inch to Feet

2. Feet to Meter 4. Meter to Feet

Solution::

#!/bin/bash

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "#### Unit convertor####\*"

echo "1.: Feet to Inch \*"

echo "2.: Feet to Meter \*"

echo "3.: Inch to Feet \*"

echo "4.: Meter to Feet \*"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

read -p "Enter option [1-4]: " key

case $key in

1)

echo "\*\* Feet to Inch \*\*"

read -p "enter feet value to convert in inch " val

convert=$(($val\*12));

echo $val "feet is eqals to" $convert "inch"

;;

3)

echo "\*\* Inch to Feet \*\*"

read -p "enter inch value to convert in feet " val

convert=$(($val/12));

echo $val "inch is eqals to" $convert "feet"

;;

2)

echo "\*\* Feet to Meter \*\*"

read -p "enter feet value to convert in meter" val

convert=$(($val/3.281));

echo $val "feet is eqals to" $convert "inch"

;;

4)

echo "\*\* Meter to Feet \*\*"

read -p "enter meter value to convert in feet " val

convert=$(($val\*3.281));

echo $val "feet is eqals to" $convert "inch"

;;

\*)

echo "No conversion "

;;

esac

output::

$ ./ConvertCase.sh

+ echo '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

+ echo '#### Unit convertor####\*'

#### Unit convertor####\*

+ echo '1.: Feet to Inch \*'

1.: Feet to Inch \*

+ echo '2.: Feet to Meter \*'

2.: Feet to Meter \*

+ echo '3.: Inch to Feet \*'

3.: Inch to Feet \*

+ echo '4.: Meter to Feet \*'

4.: Meter to Feet \*

+ echo '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

+ read -p 'Enter option [1-4]: ' key

Enter option [1-4]: 1

+ case $key in

+ echo '\*\* Feet to Inch \*\*'

\*\* Feet to Inch \*\*

+ read -p 'enter feet value to convert in inch ' val

enter feet value to convert in inch 24

+ convert=288

+ echo 24 'feet is eqals to' 288 inch

24 feet is eqals to 288 inch