**DAY 6 PRACTICE PROBLEMS**

**BY PRAMOD KUMAR**

**FOR LOOP :-**

**QUES.1 Table of 2^n**

#! /bin/bash

read -p "Enter the value of n: " n

for (( counter=1; counter<=n; counter++ ))

do

result=$(( 2\*\*$counter ))

echo "2^"$counter"="$result

done

OP:-

Enter the value of n: 6

2^1=2

2^2=4

2^3=8

2^4=16

2^5=32

2^6=64

**Ques.2 Print nth Harmonic Number**

#! /bin/bash

read -p "Enter a number: " number

harmonic=1

for (( counter=2; counter<=$number; counter++ ))

do

harmonic=`echo "1" $counter $harmonic | awk '{print$1/$2" "$3}' | awk '{print$1+$2}'`

done

echo $number" Harmonic number is "$harmonic

OP:-

Enter a number: 8

8 Harmonic number is 2.71786

**Ques.3 Input number is Prime or not.**

#! /bin/bash -x

read -p "Enter the number: " isPrime

flag=0

for (( counter=2; counter<=$(( $isPrime/2 )) ; counter++ ))

do

if [ $(( $isPrime%$counter )) -eq 0 ]

then

flag=1

break

fi

done

if [ $flag == 0 ]

then

echo $isPrime" is a Prime Number"

else

echo $isPrime" is not a Prime Number"

fi

OP:-

Enter the number: 15

15 is not a Prime Number

**Ques.4 Print range of Prime Numbers.**

#! /bin/bash

read -p "Enter the lower range: " lRange

read -p "Enter the upper range: " uRange

for (( counter=$lRange; counter<=$uRange ; counter++ ))

do

count=0

for (( i=2; i<=$(( $counter/2 )); i++ ))

do

if [ $(( $counter%$i )) -eq 0 ]

then

(( count++ ))

break

fi

done

if [ $count -eq 0 -a $counter != 1 ]

then

echo $counter

fi

done

OP:-

Enter the lower range: 10

Enter the upper range: 35

11

13

17

19

23

29

31

**Ques.5 Factorial of a input number n.**

#! /bin/bash

read -p "Enter the number: " number

sum=1

for (( counter=$number; counter>=1; counter-- ))

do

sum=$(( $counter\*$sum ))

done

echo "Factorial of "$number" is "$sum

OP:-

Enter the number: 5

Factorial of 5 is 120

**Ques.6 Compute factors of N using Prime Factorization.**

#! /bin/bash

read -p "Enter the number: " number

echo "Prime Factors of "$number " are: "

while [ $(( $number%2 )) -eq 0 ]

do

echo "2"

number=$(( $number/2 ))

done

for (( counter=3; $counter\*$counter<=$number; counter=$(( $counter+2 )) ))

do

while [ $(( $number%$counter )) -eq 0 ]

do

echo $counter

number=$(( $number/$counter ))

done

done

if [ $number -gt 2 ]

then

echo $number

fi

OP:-

Enter the number: 520

Prime Factors of 520 are:

2

2

2

5

13

**WHILE LOOP:**

**Ques.1 Table of 2^n till 256**

#! /bin/bash

read -p "Enter the value of n: " number

counter=1

while [ $counter -lt $number ] && [ $(( 2\*\*$counter )) -le 256 ]

do

result=$(( 2\*\*$counter ))

echo "2^"$counter"="$result

(( counter++ ))

done

OP:-

Enter the value of n: 12

2^1=2

2^2=4

2^3=8

2^4=16

2^5=32

2^6=64

2^7=128

2^8=256

**Ques.2 Find Magic Number**

#! /bin/bash

read -p "Enter a number b/w 1 to 100: " guess

low=0

high=100

while [ $low -le $high ]

do

mid=$(( (( $low+$high ))/2 ))

echo "1.Type 1 if the guessed number is Greater than "$mid

echo "2.Type 2 if the guessed number is Less than "$mid

read -p "Choose one of the above: " finder

if [ $guess -eq $mid ]

then

"Guessed number is "$mid

break

elif [ $guess -lt $mid ]

then

high=$(( $mid-1 ))

else

low=$(( $mid+1 ))

fi

done

OP:-

Enter a number b/w 1 to 100: 56

1.Type 1 if the guessed number is Greater than 50

2.Type 2 if the guessed number is Less than 50

Choose one of the above: 1

1.Type 1 if the guessed number is Greater than 75

2.Type 2 if the guessed number is Less than 75

Choose one of the above: 2

1.Type 1 if the guessed number is Greater than 62

2.Type 2 if the guessed number is Less than 62

Choose one of the above: 2

1.Type 1 if the guessed number is Greater than 56

2.Type 2 if the guessed number is Less than 56

Choose one of the above: 1

Guessed number is 56

**Ques.3 Flip coin Head or Tail wins 11 times.**

#! /bin/bash

headCount=0

tailCount=0

while [ $headCount -lt 11 ] && [ $tailCount -lt 11 ]

do

toss=$(( RANDOM%2 ))

if [ $toss -eq 0 ]

then

headCount=$(( $headCount+1 ))

else

(( tailCount++ ))

fi

done

if [ $headCount -gt $tailCount ]

then

echo "Head Wins"

echo "Score"

echo "Head= "$headCount" Tail= "$tailCount

else

echo "Tail Wins"

echo "Score"

echo "Head= "$headCount" Tail= "$tailCount

fi

OP:-

Head Wins

Score

Head= 11 Tail= 7

**Ques.4 Gambling Problem**

#! /bin/bash

currAmount=100

bet=1

broke=0

goal=200

noOfBets=0

winCount=0

headValue=0

tailValue=1

while [ $currAmount -gt $broke ] && [ $currAmount -lt $goal ]

do

gamble=$(( RANDOM%2 ))

if [ $gamble -eq $headValue ]

then

(( winCount++ )) && currAmount=$(( $currAmount+$bet ))

else

currAmount=$(( $currAmount-$bet ))

fi

(( noOfBets++ ))

done

echo "GAMBLING RECORD"

if [ $currAmount -eq 0 ]

then

echo "Lost"

else

echo "Won"

fi

echo "Amount= "$currAmount

echo "No of Times Won= "$winCount

echo "No of Bets Made= "$noOfBets

OP:-

GAMBLING RECORD

Won

Amount= 200

No of Times Won= 3075

No of Bets Made= 6049

**Functions:-**

**Ques1: degF <-> degC**

#! /bin/bash

function toFahrenheit() {

degF=`echo $1 | awk '{print$1\*(9/5)+32}'`

echo $degF

}

function toCelsius() {

degC=`echo $1 | awk '{print$1-32}' | awk '{print$1\*(5/9)}'`

echo $degC

}

echo "1.Convert temperature from Fahrenheit to Celsius"

echo "2.Convert temperature from Celsius to Fahrenheit"

read -p "Enter your choice (1, 2): " choice

case ${choice} in

1)

read -p "Enter temperature in Fahrenheit: " temp

celsius=`toCelsius $temp`

echo $temp" F = "$celsius " C"

;;

2)

read -p "Enter temperature in Celsius: " temp

fahrenheit=`toFahrenheit $temp`

echo $temp " C = "$fahrenheit " F"

;;

\*)

echo "Invalid Option"

;;

esac

OP:-

1.Convert temperature from Fahrenheit to Celsius

2.Convert temperature from Celsius to Fahrenheit

Enter your choice (1, 2): 1

Enter temperature in Fahrenheit: 98.5

98.5 F = 36.9444 C

**Ques2: Check a number is Palindrome or not.**

#! /bin/bash

function reverseDigits() {

rev\_num=0

num=$1

while [ $num -gt 0 ]

do

rev\_num=$(( $rev\_num\*10+$num%10 ))

num=$(( $num/10 ))

done

echo $rev\_num

}

function isPalindrome() {

rev\_n=`reverseDigits $1`

if [ $rev\_n -eq $1 ]

then

echo "1"

else

echo "0"

fi

}

read -p "Enter a number: " number

check=`isPalindrome $number`

if [ $check -eq 1 ]

then

echo "Number is Palindrome"

else

echo "Number is not Palindrome"

fi

OP:-

Enter a number: 1566

Number is not Palindrome

**Ques3: Check PalPrime**

#! /bin/bash

function reverseDigits() {

rev\_num=0

num=$1

while [ $num -gt 0 ]

do

rev\_num=$(( $rev\_num\*10+$num%10 ))

num=$(( $num/10 ))

done

echo $rev\_num

}

function isPalindrome() {

rev\_n=`reverseDigits $1`

primePalin=`isPrime $rev\_n`

if [ $primePalin -eq 1 ]

then

echo "1"

else

echo "0"

fi

}

function isPrime() {

flag=0

prime=$1

for (( counter=2; counter<=$(( $prime/2 )) ; counter++ ))

do

if [ $(( $prime%$counter )) -eq 0 ]

then

flag=1

break

fi

done

if [ $flag == 0 ]

then

echo "1"

else

echo "0"

fi

}

read -p "Enter a number: " number

primeCheck=`isPrime $number`

if [ $primeCheck -eq 1 ]

then

echo $number" Number is Prime"

checkPalindrome=`isPalindrome $number`

if [ $checkPalindrome -eq 1 ]

then

echo $number " Palindrome is also Prime"

else

echo $number " Palindrome is not Prime"

fi

else

echo $number" Number is not Prime"

fi

OP:-

Enter a number: 151

151 Number is Prime

151 Palindrome is also Prime