**DAY6 Assignment**

**Repetition Practice Problems with for loop**

1. Write a program that takes a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n.

#!/bin/bash

num=2

read -p "PLease enter power of 2: " pow

for((count=1;count<=$pow;count++))

do

echo "$(($num\*\*$count))"

done

2. Write a program that takes a command-line argument n and prints the nth harmonic number. Harmonic Number is of the form

echo -n "Please enter the number: "

read Number

Fact=1

for((i=1;i<=Number;i++))

do

Fact=$[$Fact\*$i]

done

echo "Factorial of $Number is $Fact"

3. Write a program that takes a input and determines if the number is a prime.

read -p "Please enter the number: " num

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

do

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

then

echo "$num is not prime"

exit

fi

done

echo "$num is prime number"

4. Extend the program to take a range of number as input and output the Prime Numbers in that range.

echo "Enter the range : "

read range

count=0

for ((n=2;n<=$range;n++))

do

flag=1

for ((i=2;i<$n;i++))

do

echo $n

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

then

flag=0

break;

fi

done

if [$flag -eq 1]

then

echo "$n is a prime number"

count=`expr $count + 1`

fi

done

echo "Total prime number upto $range are $count"

5. Write a program that computes a factorial of a number taken as input. 5 Factorial – 5! = 1 \* 2 \* 3 \* 4 \* 5

read -p "Enter a number to find it's factorial: " num

fact=1

for((i=2;i<=num;i++))

do

fact=$((fact\*i))

done

echo $fact

6. Write a program to compute Factors of a number N using prime factorization method.

Logic -> Traverse till i\*i <= N instead of i <= N for efficiency.

O/P -> Print the prime factors of number N.

echo "enter an integer:"

read input

i=2

count=0

flag=0

for ((i;i<$input;));do

if [ `expr $input % $i` -eq 0 ];then

factor=$i

for ((j=2;j<=`expr $factor / 2`;));do

flag=0

if [ `expr $factor % $j` -eq 0 ];then

flag=1

break

fi

j=`expr $j + 1`

done

if [ $flag -eq 0 ];then

echo $factor

count=1

fi

fi

i=`expr $i + 1`

done

if [ $count -eq 0 ];then

echo "no prime factors found except 1 and $input"

fi

**Repetition Practice Problems with while loop**

1. Write a program that takes a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n till 256 is reached..

read -p "Please enter power of 2; " pow

sum=$((2\*\*$pow))

num=2

count=1

while [ $num -lt $sum ]

do

num=$((2\*\*$count))

count=$(($count+1))

echo $num

if [ $num -eq 256 ]

then

break

fi

done

2. Find the Magic Number

a. Ask the user to think of a number n between 1 to 100

b. Then check with the user if the number is less then n/2 or greater

c. Repeat till the Magic Number is reached..

echo "---------Welcome to Magic Number----------"

echo "Think of any number between 1 to 100."

MAX\_LIMIT=100

MIN\_LIMIT=0

middle=0

choice=0

while [[(($choice -le 3))]]

do

middle=$(( ($MIN\_LIMIT+$MAX\_LIMIT)/2 ))

echo "Enter 1 : If your number greater than : $middle"

echo "Enter 2 : If your number less han : $middle"

echo "Enter 3 : If your number equal to : $middle"

read -p "Please select the above options : " choice

case $choice in

1)

MIN\_LIMIT=$(($middle+1))

;;

2)

MAX\_LIMIT=$(($middle-1))

;;

3)

echo "I gussed your number $middle"

exit

;;

esac

done

3. Extend the Flip Coin problem till either Heads or Tails wins 11 times.

flip=$(($(($RANDOM%21))%2))

head=0

tail=0

if [ $flip -eq 1 ]

then

((head++))

else

((tail++))

fi

while [ $head -ge 11 ]

do

echo "Heads"

done

4. Write a Program where a gambler starts with Rs 100 and places Re 1 bet until he/she goes broke i.e. no more money to gamble or reaches the goal of Rs 200. Keeps track of number of times won and number of bets made.

echo "Gambler program"

money=100

won=0

bet=0

min=1

max=2

while [[ (($money -ge 0)) && (($money -ne 200)) ]]

do

val=$(( $((RANDOM % $((($max -$min)+1)) ))+$min))

if [ $val -eq 1 ]

then

((money++))

((won++))

else

((money--))

fi

((bet++))

done

echo "$won number of times you won."

echo "$bet number of times bet."

echo "Money $money."

**Functions Practice Problems**

1. Help user find degF or degC based on their Conversion Selection. Use Case Statement and ensure that the inputs are within the Freezing Point (0 °C / 32 °F ) and the Boiling Point of Water ( 100 °C / 212 °F )

a. degF = (degC \* 9/5) + 32

b. degC = (degF – 32) \* 5/9

echo "Temparature conversion"

echo "1. Convert temparature Celsius to Faranhit"

echo "2. Convert temparature Faranhit to Celsius"

read -p "Enter your choice: " opt

function toCel()

{

read -p"Enter temparature in degF :" far

tc=$(((($far-32)\*5)/9))

echo "$far F = $tc C"

}

function toFar()

{

read -p"Enter temparature in degC :" cel

tf=$(((($cel\*9)/5)+32))

echo "$cel C = $tf F"

}

if [ $opt -eq 1 ]

then

toFar

elif [ $opt -eq 2 ]

then

toCel

else

echo "Enter valid option"

fi

2. Write a function to check if the two numbers are Palindromes

echo "Palindrome"

read -p "Enter first number: " num1

read -p "Enter second number: " num2

function isPalindrome(){

temp1=$num1

temp2=$num2

rev1=0

rev2=0

while [[ (( $num1 -gt 0 )) && (( $num2 -gt 0 )) ]]

do

rem1=$(( $num1%10 ))

rem2=$(( $num2%10 ))

num1=$(( $num1/10 ))

num2=$(( $num2/10 ))

rev1=$(( $rev1\*10+$rem1 ))

rev2=$(( $rev2\*10+$rem2 ))

done

if [ $temp1 -eq $rev1 ]

then

echo "$temp1 is a palindrome number"

else

echo "$temp2 is not a palindrome number"

fi

if [ $temp2 -eq $rev2 ]

then

echo "$temp2 is a palindrome number"

else

echo "$temp2 is not a palindrome number"

fi

}

isPalindrome

3. Take a number from user and check if the number is a Prime then show that its palindrome is also prime

a. Write function check if number is Prime

b. Write function to get the Palindrome.

c. Check if the Palindrome number is also prime

read -p "Enter the number: " num

#Write function to check if the number is prime.

function isPrime()

{

flag=1

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

do

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

then

flag=0

break

else

flag=1

fi

done

if [ $flag -eq 1 ]

then

echo "$num is prime number"

else

echo "$num is not prime number"

fi

}

isPrime

prime="$( isPrime $num)"

echo "Is Prime Number or not"

#Write the function to get palindrome.

function isPallindrome()

{

temp=$num

rev=0

while [ $num -gt 0 ]

do

rem=$(( $num%10 ))

num=$(( $num/10 ))

rev=$(( $rev\*10+$rem ))

done

if [ $temp -eq $rev ]

then

echo "$num is a palindrome number"

else

echo "$num is not a palindrome number"

fi

}

isPallindrome

pallNum="$( isPallindrome $num)"