**DAY 6 PRACTICE PROBLEMS**

Q1.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.

Q3. Write a Program that takes a input and determines a number is prime or not.

Ans: #!/bin/bash

read -p "Enter a number to check:" num

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

do

# ans=$(( num%i ))

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

then

echo "$num is not a prime number."

Else

echo "$num is a prime number."

Fi

done

Q4 Extend the program to take Range as input and display number of prime numbers in Range.

Ans:

#!/bin/bash

Read -p “Enter the highest number:” num

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

do

output=$(( $I/2 % 2))

if [ $output -ne 0 ]

then

echo "Prime numbers are: $i"

Fi

Done

Q5.Write a program to compute factorial of a number taken as input.

Ans:

#!/bin/bash

read -p "enter a number for factorial:" num

fact=1

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

do

fact=$(( $fact\*$i ))

done

echo "$( getfactorial $fact)"

Q6.Write a program to compute factors of a number N using prime factorisation method.

Ans:

#! /bin/bash -x

read -p "Enter an integer greater than one:" num

i=2

count=0

flag=0

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

if [ `expr $num % $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 $num"

fi

Q7.Write a program that takes 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.

Ans: # /bin/bash -x

i=2

while [ $i -le 256 ]

do

echo $i

i=$((2\*i))

done

Q8.Find the magic number

1. Ask the user to think a number n between 1 to 100
2. Then check with the user if the number is less than n/2 or greater.
3. Repeat till the magic number is reached.

Ans:

Q9. Extend cointoss problem till heads or tails win 11 times.

Ans:

#! /bin/bash

MAX\_HEADS=11

MAX\_TAILS=11

totalHeads=0

totalTails=0

while [ $totalHeads -le 11 ] && [ $totalTails -le 11 ]

do

echo "Welcome to CoinToss"

coinToss=$(( $RANDOM %2 + 1 ))

echo "Choose an option 1.Head 2.Tail"

read -p "Enter your Choice:" choice

if [ $choice -eq $coinToss ]

then

echo " heads"

else

echo "tails"

fi

done

Q11. Program to convert temperature from farenheight to celcious and vice-versa.

Ans:

function myfunct(){

echo $1

}

**echo** "1. Convert Celsius temperature into Fahrenheit"

**echo** "2. Convert Fahrenheit temperatures into Celsius"

**echo** -n "Select your choice (1-2) : "

**read** choice

**if** **[** $choice -eq 1 **]then**

**echo** -n "Enter temperature (C) : "

**read** tc

*# formula Tf=(9/5)\*Tc+32*

tf=$**(echo** "scale=2;((9/5) \* $tc) + 32" **|bc)**

**echo** "$tc C = $tf F"**elif** **[** $choice -eq 2 **]then**

**echo** -n "Enter temperature (F) : "

**read** tf

*# formula Tc=(5/9)\*(Tf-32)*

tc=$**(echo** "scale=2;(5/9)\*($tf-32)"**|bc)**

**echo** "$tf = $tc"**else**

**echo** "Please select 1 or 2 only"

**exit** 1

**fi**

Q12. Write a function to check if two numbers are palindromes.

Ans:

#! /bin/bash -x

read -p "Enter first number to check:" n

#n=12321

read -p "Enter second number to check:" rev

rev=$( echo $n | rev)

if [ $n -eq $rev ];

then

echo "Number is palindrome"

else

echo "Number is not palindrome"

fi

Q13.(a) Write a function to check a number is prime or not

Ans:

function prime()

{

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

do

if [ `expr $num % $i` == 0 ]

then

echo $num is not prime

exit

fi

done

echo $num is a prime number

}

read num;

prime "$num";

(b) Write a function to check the number is pelindrome or not.

Ans:

Function checkpalindrome()

{

Local s =$1

For n in $s;

Do

While [ $n -gt 0 ]

rev=$( echo $n | rev)

if [ $n -eq $rev ];

then

echo "Number is palindrome"

else

echo "Number is not palindrome"

fi

}