**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**

read -p " Enter the number: " n

for ((i=0;i<=n;i++))

do

echo $((2\*\*i))

done

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

read -p " Enter the number: " n

sum=0

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

do

div=$( awk ' BEGIN { print '1' / '$i' } ' )

sum=$( awk ' BEGIN { print '$sum' + '$div' } ' )

done

echo " Harmonic value of $n is $sum: "

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

read -p " Enter the number: " n

counter=0

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

do

a=$(($n % $i))

if [ $a -eq 0 ]

then

((counter++))

fi

done

if [ $counter -eq 2 ]

then

echo " prime number "

else

echo " is not prime number "

fi

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

read -p " enter the number between 1 to 200 : " num

div=0

if [ $num -lt 200 ]

then

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

do

a=$(( $num % $i ))

if [ $a -eq 0 ]

then

(( div++ ))

fi

done

if [ $div -eq 2 ]

then

echo " prime number "

else

echo " not a prime number "

fi

else

echo " number between 1 to 200 "

fi

**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 the number: " n

a=1

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

do

a=$(( $a \* $i ))

done

echo "$a"

**6Write 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**

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

do

read -p "enter the number:" num

if [ $((num%$i)) == 0 ]

then

echo "$i"

num=$((num/$i))

fi

done

**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**

read -p "Enter the number:" n

a=0

while [ $a -lt 9 ]

do

echo $((2\*\*$a))

((a++))

done

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

a=0

while [ $a -lt 11 ]

do

coin=$((RANDOM%2))

if [ $coin -eq 0 ]

then

echo " this is head "

else

echo " this is tail "

fi

((a++))

done

function

1>read -p " Enter the value: " a

read -p " select 1. convert F to C 2. convert C to F: " choice

function conversion() {

case $choice in

1) echo convert F to C:$(( ($a - 32) \* 5/9 )) ;;

2) echo convert C to F:$(( ($a \* 9/5) + 32 )) ;;

\*) echo " no conversion " ;;

esac

}

resultconversion= "$( conversion )"

2.read -p " enter the number: " num

a=$num

reverse=""

reminder=0

function palindromenumber() {

while [ $num -gt 0 ]

do

reminder=$(( $num % 10 ))

num=$(( $num / 10 ))

reverse=$( echo ${reverse}${reminder} )

echo "$a + a"

done

echo "$reverse + reverse "

if [ $a == $reverse ]

then

echo " the number is palindrome "

else

echo " the number is not palindrome "

fi

}

palindromenumber

2.read -p " enter the number: " num

a=$num

reverse=""

reminder=0

function palindromenumber() {

while [ $num -gt 0 ]

do

reminder=$(( $num % 10 ))

num=$(( $num / 10 ))

reverse=$( echo ${reverse}${reminder} )

echo "$a + a"

done

echo "$reverse + reverse "

if [ $a == $reverse ]

then

echo " the number is palindrome "

else

echo " the number is not palindrome "

fi

}

palindromenumber

3.read -p "enter the number:" n

counter=0

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

do

a=$(($n % $i))

if [ $a -eq 0 ]

then

((counter++))

fi

done

if [ $counter -eq 2 ]

then

echo " prime number "

else

echo " is not prime number "

fi

read -p " enter the number: " num

a=$num

reverse=""

reminder=0

function palindromenumber() {

while [ $num -gt 0 ]

do

reminder=$(( $num % 10 ))

num=$(( $num / 10 ))

reverse=$( echo ${reverse}${reminder} )

echo "$a + a"

done

echo "$reverse + reverse "

if [ $a == $reverse ]

then

echo " the number is palindrome "

else

echo " the number is not palindrome "

fi

counter=0

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

do

x=$(($num % $i))

if [ $x -eq 0 ]

then

((counter++))

fi

done

if [ $counter -eq 2 ]

then

echo " palindrome number is prime number "

else

echo "palindrome number is not prime number "

fi

}

palindromenumber