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

echo $((RANDOM%10))

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

echo $((RANDOM%6 +1))

**3.Add two Random Dice Number and Print the Result**

dice1=$((RANDOM%6))

echo First random dice number : $dice1

dice2=$((RANDOM%6))

echo Second random dice number : $dice2

sum=$((dice1+dice2))

echo Addition of two random dice number : $sum

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

**sum and the average**

echo "Two Digit random number... "

for((cnt=1; cnt<=5; cnt++))

do

twoDigitNum=$(((RANDOM%90) + 10))

echo $twoDigitNum

sum=`expr $sum + $twoDigitNum`

done

echo "Total Sum of random number..." $sum

avg=`expr $sum / 5`

echo "Total average of random number..." $avg

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

**and the maximum value**

echo "three Digit random number... "

threeDigitNum1=$(((RANDOM%900) + 100))

threeDigitNum2=$(((RANDOM%900) + 100))

threeDigitNum3=$(((RANDOM%900) + 100))

threeDigitNum4=$(((RANDOM%900) + 100))

threeDigitNum5=$(((RANDOM%900) + 100))

max=$threeDigitNum1

min=$threeDigitNum1

if [ $threeDigitNum2 -gt $max ]

then

max=$threeDigitNum2

fi

if [ $threeDigitNum2 -lt $min ]

then

min=$threeDigitNum2

fi

if [ $threeDigitNum3 -gt $max ]

then

max=$threeDigitNum3

fi

if [ $threeDigitNum3 -lt $min ]

then

min=$threeDigitNum3

fi

if [ $threeDigitNum4 -gt $max ]

then

max=$threeDigitNum4

fi

if [ $threeDigitNum4 -lt $min ]

then

min=$threeDigitNum4

fi

if [ $threeDigitNum5 -gt $max ]

then

max=$threeDigitNum5

fi

if [ $threeDigitNum5 -lt $min ]

then

min=$threeDigitNum5

fi

echo "Maximum value...." $max

echo "Minimum value...." $min

**6.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.**

read -p "Enter Date : " date

read -p "Enter Month : " month

if [ $month -ge 3 ] && [ $month -le 6 ]

then

echo "True.."

else

echo "False.."

fi

**7.** **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.**

read -p "Enter four Digit Year.." year

if [ `expr $year % 4` -eq 0 ] && [ `expr $year % 100` -ne 0 ] || [ `expr $year % 400` -eq 0 ]

then

echo $year " is a leap year.."

else

echo $year "is not a leap year.."

fi

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

coinFlip=$((RANDOM%2))

if [ $coinFlip -eq 1 ]

then

echo "heads"

else

echo "tails"

fi

**9. 1ft = 12 in then 42 in = ? ft**

read -p "Enter the number of inches.." inch

feet=`expr $inch / 12`

echo $inch "=" $feet

**10.** **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 number.." n

i=0

powerOfTwo=1

while [ $i -le $n ]

do

echo $i" "$powerOfTwo

powerOfTwo=$((2 \* $powerOfTwo))

i=$(($i+1))

done

**11.** **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

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

do

if [ $i -eq 1 ]

then

addSeries="(1/$i)"

else

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

do

if [ $i -eq $j ]

then

addSeries="${addSeries}+(1/${i})"

fi

done

fi

done

series=$addSeries;

echo "Harmonic series are..."$series

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

read -p "Enter number..." num

flag=0

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

do

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

then

echo $num "is not a prime number.."

flag=1

fi

done

if [ $flag -eq 0 ]

then

echo $num "is a prime number.."

fi

**13.** **Extend the program to take a range of number as input and output the Prime**

**Numbers in that range.**

read -p "Enter the Start number.." start

read -p "Enter the End number.." end

for ((i=$start;i<=$end;i++))

do

count=0

for((j=1;j<=$i;j++))

do

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

then

count=$(($count+1))

fi

done

if [ $count -eq 2 ]

then

echo $i

fi

done

**14.** **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" num

fact=1

i=1

while [ $i -le $num ]

do

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

i=$(($i+1))

done

echo $fact

**15.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 "Enter number.." n

i=0

powerOfTwo=1

while [ $i -le 8 ]

do

echo $i" "$powerOfTwo

powerOfTwo=$((2 \* $powerOfTwo))

i=$(($i+1))

done

**16.** **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..**

read -p "Enter any number between 1 to 100 " n

i=1

while [ $i -le $n ]

do

if [ $i -lt $(($n/2)) ]

then

echo "Number is less .."$i

else

echo "Number is greater .."$i

fi

i=$(($i+1))

done

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

i=1

while [ $i -le 11 ]

do

flip=$(($(($RANDOM%10))%2))

if [ $flip -eq 1 ]

then

echo "heads"

else

echo "tails"

fi

i=$(($i+1))

done

**18.** **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**

function calculateTempConversion(){

read -p "Choose type of conversion \n 1.Fahrenheit to Celsius \n 2.Celsius to >

case $num in

1)

read -p "Enter Fahrenheit temperature.." f

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

echo "Celsius temperature is = "$c

;;

2)

read -p "Enter Celsius temperature.." c

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

echo "Fahrenheit temperature is = "$f

;;

esac

}

calculateTempConversion

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

function isPalindrome(){

read -p "Enter any number.." num

temp=$num

while [ $num -gt 0 ]

do

s=$(( $num % 10 ))

num=$(( $num / 10 ))

rev=$( echo ${rev}${s} )

done

if [ $temp -eq $rev ]

then

echo "Number is Palindrome"

else

echo "Not Palindrome"

fi

}

isPalindrome

**20.** **Write a program in the following steps**

**a. Generates 10 Random 3 Digit number.**

**b. Store this random numbers into a array.**

**c. Then find the 2nd largest and the 2nd smallest element with sorting the array.**

for ((a=1; a<=10;a++))

do

arr[$a]=$((RANDOM%900 +100))

done

echo ${arr[@]}

size=${#arr[@]}

for ((i=1; i<=$size ;i++))

# 1 3 1++

do

for (( j=$i+1; j<=$size ;j++))

# 2 2<=3 2++

do

if [ ${arr[i]} -gt ${arr[j]} ]

then

temp=${arr[i]};

echo "temp.."$temp

arr[i]=${arr[j]};

echo "arr[i]..."${arr[i]}

arr[j]=$temp;

echo "arr[j].."${arr[j]}

fi

done

done

echo " Array.." ${arr[@]}

echo "Second largest number..."${arr[$size-1]}

echo "Second smallest number..."${arr[2]}

**21.** **Write a Program to show Sum of three Integer adds to ZERO**

read -p "Enter the size of array.. " size

for((a=0;$a<$size;a++))

do

read -p "Enter array element .." array

arr[$a]=$array

done

for ((i=0; $i<$size; i++))

do

for ((j=$i+1; $j<$size; j++))

do

for ((l=$j+1; $l<$size; l++))

do

iVal=${arr[i]}

jVal=${arr[j]}

lVal=${arr[l]}

#if [ "${arr[i]} + ${arr[j]} + ${arr[l]}" ]

if [ $((iVal+(jVal+lVal))) -eq 0 ]

then

echo "Found 3 elements whose sum is = 0"

echo "Elements are " $iVal " " $jVal " " $lVal

fi

done

done

done

**22.Move files from one folder to the respective folders.**

for files in \*.txt

do

folderName=`echo $files | awk -F. '{print $1}'`;

echo $files;

echo $folderName;

printf "\n";

if [ -d $folderName ]

then

rm -r $folderName;

fi

mkdir $folderName;

cp $files $folderName/$files;

done

**23.Append current date to all log files name which has extention .log.1 from a folder**

for file in \*.log.1

do

echo $file;

fileName=`echo $file | awk -F. '{print $1}'`;

echo fileName;

extName=`echo $file | awk -F. '{print $2}'`;

echo extName;

now=$(date +"%d%m%Y");

echo $now;

nFile=$fileName-$now.$extName;

echo $nFile;

(mv $file $nFile);

done

**24.chec k if folder exist or not if its not present create it**

dir=Bckps;

if [ ! -d $dir ]

then

mkdir $dir

echo "Folder Created"

else

echo "Folder exists"

fi

**25.Execute command “Hello” and “ls” and check its execution status and print whether command executed successfully or not**

for cmds in hello ls

do

echo $cmds

$cmds

num=`echo $?`

echo $num

chkNum=0

if [ $num -eq $chkNum ];

then

echo "Command executed successfully"

else

echo "command not executed successufully"

fi

done

**26.Find a word “error” from all log files in the folder /var/log and print number of occurrence more than 0 against each file.**

for files in var/log/\*.log

do

echo $files

noOfOccur=`grep -o -i error $files | wc -w`

if [ $noOfOccur -gt 0 ]

then

echo Number of Occurence of word error is : $noOfOccur

fi

done

**27.Print Employee Name and Total Pay who has BasePay is greater than 10000**

for basePay in `cat data.csv | awk '{print $4}'`

do

echo $basePay

if [ $basePay -gt 10000 ]

then

cat data.csv | awk '{print $2} {print $7}'

fi

done

**28.what is the aggregate TotalPay of employees whose job title is captain**

cat data.csv|grep -i captain |gawk '{ sum += $7 }; END { print sum }'

**29.Print JobTitle and OverTimePay who has overTimePay is between 7000 and 10000.**

cat data.csv| awk '{if ($5 >= 7000 && $5 <= 10000) print $3 " " $5}'

**30.Print average BasePay.**

cat data.csv|awk '{ sum += $4 } END { if (NR > 0) print sum / NR }'