**# to check leap year**

puts "Enter the year you want to check"

yr = gets.chomp.to\_i

if yr % 400 == 0

puts " is a leap year"

elsif yr % 4 == 0 && yr % 100 !=0

puts " is a leap year"

else

puts " is not a leap year"

end

**#to check the number is positive or negative**

puts "Enter number: ";

num = gets.chomp.to\_i;

msg=num>0 ? "POSITIVE" : "NEGATIVE";

puts "Number is: ",msg;

**#to check the given number is even or odd**

puts "Enter number: ";

number = gets.chomp.to\_i;

Msg = (number%2==0) ? "EVEN" : "ODD";

puts "Number is: ",Msg;

**# to find largest of 3 numbers**

puts "Enter first number: ";

a = gets.chomp.to\_i;

puts "Enter second number: ";

b = gets.chomp.to\_i;

puts "Enter third number: ";

c = gets.chomp.to\_i;

largest=a>b ? a>c ?a : c : b>c ? b : c;

puts "

**# to find a factorial of anumber**

puts "Enter the number:"

num=gets.chomp.to\_i

fact=1

if (num==0)

puts "Error! Could not find the factorial of one"

else

i=1

while(i<=num)

fact=fact\*i

i+=1

end

end

puts "factorial of #{num} is #{fact}"

**# To generate multiplication table**

puts "Enter the number:"

num=gets.chomp.to\_i

for i in 1..10

mult=num\*i

puts "#{num} \* #{i} = #{mult}"

end

**# to print Fibonacci series**

first=0

second=1

nextterm=0

puts "Enter the number of terms:-"

n=gets.chomp.to\_i

puts "The first #{n} terms of Fibonacci series are:-"

c=1

while(c<=n+1)

if(c<=1)

nextterm=c

else

puts nextterm

nextterm=first+second

first=second

second=nextterm

end

c+=1

end

**# to find GCD of two numbers**

def calculateGCD(a, b)

while (a != b)

if a > b

return calculateGCD(a - b, b);

else

return calculateGCD(a, b - a);

end

end

return a;

end

print "Enter number1: ";

number1 = gets.chomp.to\_i;

print "Enter number2: ";

number2 = gets.chomp.to\_i;

result = calculateGCD(number1, number2);

print "GCD is: ",result;

**#to count the number of digits in an integer**

puts "Enter the number:"

num=gets.chomp.to\_i

temp=num

count=0

while (temp>0)

count+=1

temp=temp/10

end

puts "#{num} has #{count} digits"

**# to calculate the power if a number**

def pow(a,b)

power=1

for i in 1..b

power=power\*a

end

return power

end

puts "Enter Base:-"

base=gets.chomp.to\_i

puts "Enter exponent:-"

expo=gets.chomp.to\_i

puts "The power is #{pow(base,expo)}"

**#to check the number is prime number**

puts "Enter the number:"

num=gets.chomp.to\_i

count=0

if (num==0)

puts "0 is not prime"

else

i=2

while(i<num)

if (num%i==0)

count+=1

end

i+=1

end

end

if count>1

puts "#{num} is not a prime number"

else

puts "#{num} is a prime number"

end

**# to check amstrong number**

puts "Enter the number"

num=gets.chomp.to\_i

temp=num

sum = 0

while num!=0

rem=num%10

num=num/10

sum=sum+rem\*rem\*rem

end

if(temp==sum)

puts "The is Armstrong"

else

puts "The is not Armstrong"

end

**# program to reverse a string**

Syntax:str.reverse

Puts”prajwal”.reverse

Puts”cognitive”.reverse