Lab qn -day 1

1.Write a program to Print Fibonacci Series using recursion.

```
def recur_fibo(n):
   if n <= 1:
      return n
   else:
      return(recur_fibo(n-1) + recur_fibo(n-2))

nterms = 10

# check if the number of terms is valid
if nterms <= 0:
   print("Plese enter a positive integer")
else:
   print("Fibonacci sequence:")
   for i in range(nterms):
      print(recur_fibo(i))
```

Run Code

Output:

```
Fibonacci sequence:
0
1
1
2
3
5
8
13
21
34
```

2.Write a program to check the given no is Armstrong or not using recursive function.

```
def check_armstrong(num):
    if num == 0:
        return num
    else:
        return pow((num%10),order) + check_armstrong(num//10)

num = int(input("Enter a number to check if it is an Armstrong number or not: "))

order = len(str(num))
sum = check_armstrong(num)

if sum == int(num):
    print(num,"is an Armstrong Number.")
```

```
else:
    print(num,"is not an Armstrong Number.")
```

o/p:

Enter a number to check if it is an Armstrong number or not: 1634

1634 is an Armstrong Number.

3.Write a program to find the GCD of two numbers using recursive factorization

```
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
GCD=gcd(a,b)
print("GCD is: ")
print(GCD)
```

o/p:

Enter first number:30

Enter second number:12

GCD is:

6

4.Write a program to get the largest element of an array.

```
def largest(arr, n):

    max = arr[0]

    for i in range(1, n):
        if arr[i] > max:
            max = arr[i]
    return max
```

```
arr = [10, 324, 45, 90, 9808]
n = len(arr)
Ans = largest(arr, n)
print("Largest in given array ", Ans)
```

o/p:9808

5.Write a program to find the Factorial of a number using recursion.

```
def recur_factorial(n):
   if n == 1:
       return n
   else:
       return n*recur_factorial(n-1)

num = 7

# check if the number is negative
if num < 0:
   print(" factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   print("The factorial of", num, "is", recur_factorial(num))
```

o/p: The factorial of 7 is 5040

6. Write a program for to copy one string to another using recursion

```
def recur_factorial(n):
   if n == 1:
       return n
   else:
       return n*recur_factorial(n-1)
```

```
num = 7
if num < 0:
   print("factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   print("The factorial of", num, "is", recur_factorial(num))
```

Output:

The factorial of 7 is 5040

7.Write a program to print the reverse of a string using recursion

```
def reverse(s):
    if len(s) == 0:
        return s
    else:
        return reverse(s[1:]) + s[0]
s = "gowtham"

print("The original string is : ", end="")
print(s)
print("The reversed string(using recursion) is : ", end="")
print(reverse(s))
```

Output:

The original string is : gowtham

The reversed string(using recursion) is : mahtwog

8.Write a program to generate all the prime numbers using recursion

```
def prime(n,i=2):
    if n<=2:
        return n==2
    if n%i==0:
```

```
        return False
    if i*i>n:
        return True
    return prime(n,i+1)

def allprime(n,x=2):
    if x>n:
        return []
    if prime(x):
        return [x]+allprime(n,x+1)
    else:
        return allprime(n,x+1)

num=10
print(allprime(num))
```

output:

[2, 3, 5, 7]

9.Write a program to check a number is a prime number or not using recursion.

```
def isPrime(n, i = 2):
    if (n <= 2):
        return True if(n == 2) else False
    if (n % i == 0):
        return False
    if (i * i > n):
        return True
    return isPrime(n, i + 1)

n = 15
if (isPrime(n)):
    print("Yes")
```

```
else:
    print("No")
```

o/p:No

10.Write a program for to check whether a given String is Palindrome or not using recursion

```
def isPalindrome(s):
    return s == s[::-1]


# Driver code
s = "malayalam"
ans = isPalindrome(s)

if ans:
    print("Yes")
else:
    print("No")
```

o/p:No