LAB DAY-1(04-06-24)

1. Fibonacci Series using Recursion

```
def fibo (n):
    if n<=0:
        return 0
    elif n==1:
        return 1
    else:
        return fibo (n-1) + fibo (n-2)

for i in range(10):
    print(fibo(i), end=" ")
```

OUTPUT:
0 1 1 2 3 5 8 13 21 34

2. Check if a Number is Armstrong using Recursion

```
def count(n):
    if n == 0:
        return 0
    return 1 + count(n // 10)

def condition(n, power):
    if n == 0:
        return 0
    return (n % 10) ** power + condition(n // 10, power)

def true(n):
    power = count(n)
    return n == condition(n, power)

num = 153
print(true(num))
```

OUTPUT:
true

3. GCD of Two Numbers using Recursion

```
def gcd(x,y):
    if y== 0:
        return x
    else:
        return gcd(y, x % y)

x= 48
y= 18
print(gcd(x, y))
```

OUTPUT:
6

4. Largest Element of an Array

```
def large(arr, n):
    if n == 1:
        return arr[0]
    return max(arr[n-1], large(arr, n-1))

array = [1,10,5,7,1]
print(large(array, len(array)))
```

OUTPUT:
10

5. Factorial of a Number using Recursion

```
def fact(n):
    if n==0:
        return 1
    else:
        return n*fact(n-1)

num=5
print(fact(num))
```

OUTPUT:
120

6. Copy One String to Another using Recursion

```
def copy(str1, str2, index=0):
    if index == len(str1):
        return str2
    else:
        str2+= str1[index]
        return copy(str1, str2, index + 1)

string1= "Hello"
string2 = ""
print(copy(string1,string2))
```

OUTPUT:
Hello

7. Print the Reverse of a String using Recursion

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

string = "Hello"
print(reverse(string))
```

OUTPUT:
olleH

8. Generate All 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. Check if a Number is Prime 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)

num = 13
print(prime(num))
```

OUTPUT:

True

10. Check if a String is a Palindrome using Recursion

```
def palin(s):
    if len(s) <= 1:
        return True
    if s[0] != s[-1]:
        return False
    return palin(s[1:-1])

string = "madam"
print(palin(string))
```

OUTPUT:

True