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

```
def fib(n):
    if n<=1:
        return n
    else:
        return fib(n-1)+fib(n-2)
num=10
if num<=0:
    print("enter +ve numbers")
else:
    print("fibonacci :")
    for i in range(num):
        print(fib(i))
```

```
fibonacci :
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 armstrong(n,digit=0,temp=0):
    if digit==0:
        digit=len(str(n))
        temp=n
    if temp==0:
        return n==0
    return armstrong(n-(temp%10)**digit,digit,temp//10)
num=153
if armstrong(num):
    print("armstrong number ")
else:
    print("not armstrong number")
```

```
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
    return gcd(b, a % b)
def lcm(a, b):
    return abs(a * b) // gcd(a, b)
num1 = 24
num2 = 36
print(f"GCD of {num1} and {num2} is: {gcd(num1, num2)}")
print(f"LCM of {num1} and {num2} is: {lcm(num1, num2)}")
```

```
GCD of 24 and 36 is: 12
LCM of 24 and 36 is: 72
```

without recursion

```
import math
num1 = 24
num2 = 36
gcd = math.gcd(num1, num2)
lcm = abs(num1 * num2) // math.gcd(num1, num2)
print("GCD of", num1, "and", num2, "is:", gcd)
print("LCM of", num1, "and", num2, "is:", lcm)
```

```
GCD of 24 and 36 is: 12
LCM of 24 and 36 is: 72
```

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

```
a=[3,66,23,11,4,0,89]
largest=a[0]
for i in range(1,len(a)):
    if a[i]>largest:
        largest=a[i]
print(largest)
```

```
89
```

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

```
n=5
def fact(n):
    if n==0 or n==1:
        return 1
    else:
        return n*fact(n-1)
print(fact(n))
```

```
120
```

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

```
source="hi hi hi"
def copy(scr,index=0):
    if index==len(scr):
        return ""
    return scr[index]+copy(scr,index+1)
print(copy(source))
```

```
hi hi hi
```

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

```
def revstr(s):
    if s=="":
        return
    print(s[-1],end="")
    revstr(s[:-1])
string="hi hello"
revstr(string)
```

```
olleh ih
```

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

```
limit = 50
def is_prime(n, divisor=2):
    if n < 2:
        return False
    if divisor * divisor > n:
        return True
    if n % divisor == 0:
        return False
    return is_prime(n, divisor + 1)
def print_primes(current, limit):
    if current > limit:
        return
    if is_prime(current):
        print(current, end=' ')
```

```
    print_primes(current + 1, limit)
print_primes(2, limit)
```
```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
```

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

```
n = 29
divisor = 2
def is_prime(n, divisor):
    if n < 2:
        result = False
    elif divisor * divisor > n:
        result = True
    elif n % divisor == 0:
        result = False
    else:
        result = is_prime(n, divisor + 1)
    return result
print(is_prime(n, divisor))
```
```
True
```

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

```
def is_palindrome(s):
    if len(s) < 2:
        return True
    if s[0] != s[-1]:
        return False
    return is_palindrome(s[1:-1])
input_string = "radar"
if is_palindrome(input_string):
    print(f"{input_string} is a palindrome")
else:
    print(f"{input_string} is not a palindrome")
```
```
radar is a palindrome
```