

1. Fibonacci Series using recursion:

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

terms = int(input("Enter the number of terms: "))

if terms <= 0:
    print("Please enter a positive integer.")
else:
    print("Fibonacci Series:")
    for i in range(terms):
        print(fibonacci(i))
```

```
Enter the number of terms: 5
Fibonacci Series:
0
1
1
2
3
```

2. Check if a given number is Armstrong or not using recursion:

```
def order(num):
    return len(str(num))

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

num = int(input("Enter a number: "))

if num == isArmstrong(num):
    print(num, "is an Armstrong number.")
else:
    print(num, "is not an Armstrong number.")
```

```
Enter a number: 153
153 is an Armstrong number.
```

3. GCD of two numbers using recursive factorization:

```
def gcd(a, b):  
    if b == 0:  
        return a  
    else:  
        return gcd(b, a % b)  
  
num1 = int(input("Enter first number: "))  
num2 = int(input("Enter second number: "))  
  
print("GCD of", num1, "and", num2, "is", gcd(num1, num2))
```

```
Enter first number: 48  
Enter second number: 18  
GCD of 48 and 18 is 6
```

4. Get the largest element of an array:

```
def find_largest(arr):  
    if len(arr) == 1:  
        return arr[0]  
    else:  
        return max(arr[0], find_largest(arr[1:]))  
  
arr = [int(x) for x in input("Enter the array elements separated by space: ").split()]  
print("The largest element in the array is:", find_largest(arr))
```

```
Enter the array elements separated by space: 12 34 5 67 89  
The largest element in the array is: 89
```

5. Factorial of a number using recursion:

```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n-1)  
  
num = int(input("Enter a number: "))  
print("Factorial of", num, "is", factorial(num))
```

```
Enter a number: 5  
Factorial of 5 is 120
```

6. Copy one string to another using recursion:

```
def string_copy(src, dest, index=0):
    if index == len(src):
        return dest
    else:
        dest += src[index]
        return string_copy(src, dest, index+1)

source = input("Enter the source string: ")
destination = ""

print("Copied string:", string_copy(source, destination))
```

```
Enter the source string: Hello
Copied string: Hello
```

7. Print the reverse of a string using recursion:

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

string = input("Enter a string: ")
print("Reversed string:", reverse_string(string))
```

```
Enter a string: Python
Reversed string: nohtyP
```

8. Generate all the prime numbers using recursion:

```
def is_prime(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 is_prime(n, i + 1)

def generate_primes(start, end):
    if start <= end:
        if is_prime(start):
            print(start)
        generate_primes(start + 1, end)

start = int(input("Enter the starting number: "))
end = int(input("Enter the ending number: "))

print("Prime numbers between", start, "and", end, "are:")
generate_primes(start, end)
```

```
Enter the starting number: 10
Enter the ending number: 20
Prime numbers between 10 and 20 are:
11
13
17
19
```

9. Check if a number is prime or not using recursion

```
def is_prime(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 is_prime(n, i + 1)

num = int(input("Enter a number: "))

if is_prime(num):
    print(num, "is a prime number.")
else:
    print(num, "is not a prime number.")
```

```
Enter a number: 29
29 is a prime number.
```

10. Check whether a given string is palindrome or not using recursion:

```
def is_palindrome(s):  
    if len(s) <= 1:  
        return True  
    if s[0] != s[-1]:  
        return False  
    return is_palindrome(s[1:-1])  
  
string = input("Enter a string: ")  
  
if is_palindrome(string):  
    print("The string is a palindrome.")  
else:  
    print("The string is not a palindrome.")
```

```
Enter a string: radar  
The string is a palindrome.
```