**1. Print numbers from N to 1 using recursion**

java

CopyEdit

void printNto1(int n) {

if (n == 0) return; // Base case

System.out.print(n + " ");

printNto1(n - 1); // Recursive call

}

**Example:**  
Input: printNto1(5)  
Output: 5 4 3 2 1

**2. Print numbers from 1 to N using recursion**

java

CopyEdit

void print1toN(int n) {

if (n == 0) return; // Base case

print1toN(n - 1); // Recursive call

System.out.print(n + " ");

}

**Example:**  
Input: print1toN(5)  
Output: 1 2 3 4 5

**3. Sum of first N natural numbers using recursion**

java

CopyEdit

int sumN(int n) {

if (n == 0) return 0; // Base case

return n + sumN(n - 1); // Recursive call

}

**Example:**  
Input: sumN(5)  
Output: 15 (5+4+3+2+1)

**4. Factorial of a number using recursion**

java

CopyEdit

int factorial(int n) {

if (n == 0 || n == 1) return 1; // Base case

return n \* factorial(n - 1); // Recursive call

}

**Example:**  
Input: factorial(5)  
Output: 120 (5! = 5 × 4 × 3 × 2 × 1)

**5. Power function (a^b) using recursion**

java

CopyEdit

int power(int a, int b) {

if (b == 0) return 1; // Base case: a^0 = 1

return a \* power(a, b - 1); // Recursive call

}

**Example:**  
Input: power(2, 3)  
Output: 8 (2³ = 2 × 2 × 2)

**6. Check if a string is palindrome using recursion**

java

CopyEdit

boolean isPalindrome(String str, int start, int end) {

if (start >= end) return true; // Base case

if (str.charAt(start) != str.charAt(end)) return false;

return isPalindrome(str, start + 1, end - 1); // Recursive call

}

**Example:**  
Input: "madam" → isPalindrome("madam", 0, 4)  
Output: true

**7. Find the GCD (Greatest Common Divisor) using recursion**

java

CopyEdit

int gcd(int a, int b) {

if (b == 0) return a; // Base case

return gcd(b, a % b); // Recursive call

}

**Example:**  
Input: gcd(24, 36)  
Output: 12

**8. Find the LCM (Least Common Multiple) using recursion**

java

CopyEdit

int lcm(int a, int b) {

return (a \* b) / gcd(a, b); // Using GCD formula

}

**Example:**  
Input: lcm(4, 6)  
Output: 12

**9. Compute nth Fibonacci number using recursion**

java

CopyEdit

int fibonacci(int n) {

if (n <= 1) return n; // Base case

return fibonacci(n - 1) + fibonacci(n - 2); // Recursive calls

}

**Example:**  
Input: fibonacci(5)  
Output: 5 (0, 1, 1, 2, 3, **5**)

**10. Sum of digits of a number using recursion**

java

CopyEdit

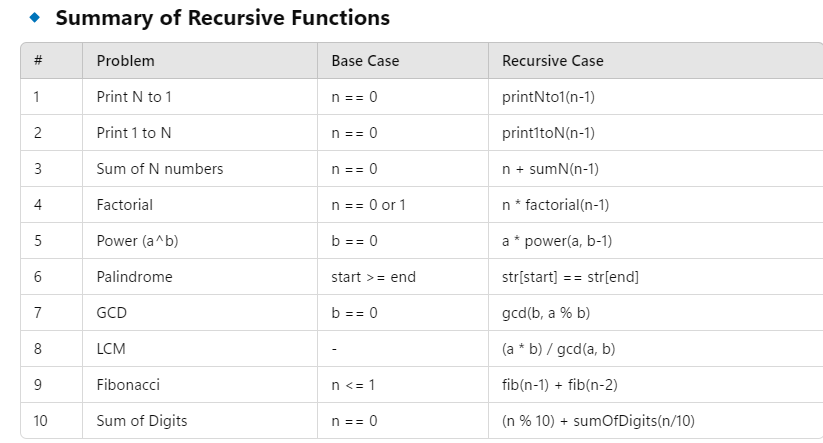
int sumOfDigits(int n) {

if (n == 0) return 0; // Base case

return (n % 10) + sumOfDigits(n / 10); // Recursive call

}

**Example:**  
Input: sumOfDigits(1234)  
Output: 10 (1+2+3+4)

****