Recursion

Simulation of a recursive function:
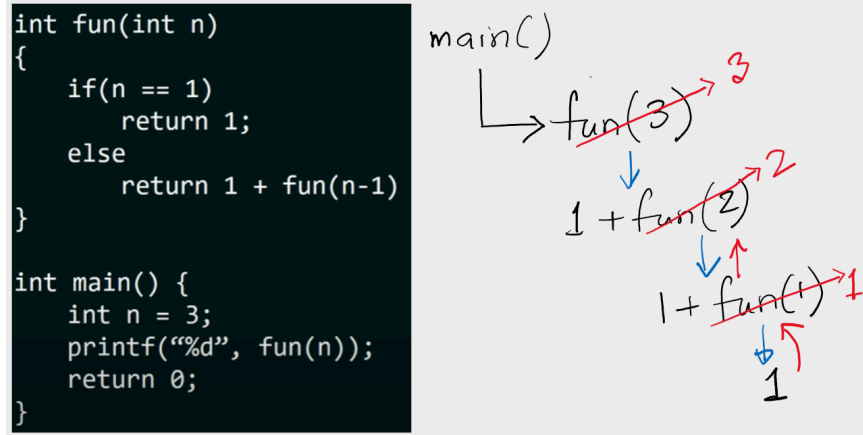
How to write recursive functions [link]

1. Divide the problem into smaller subproblems. Identify the recursive formula.
2. Specify the base condition to stop the recursion

Divide the problem into smaller subproblems.

Fact (1) = 1
Fact (2) = 2. 1 = 2. Fact (1)
Fact (3) = 3. 2. 1 = 3. Fact (2)
Fact (4) = 4. 3. 2. 1 = 4. Fact (3)
...
Fact (n) = n. Fact(n-1)

Specify the base condition to stop the recursion

The base case returns a value without making any subsequent recursive calls. For example: Fact (1) = 1

Function rec:
   if base condition:
    ...
   else
   ... recursive formula
	1. **Function** Fact(n):
	2. **if** n==1 **then**
	3. **return** 1
	4. **else**
	5. **return** n*Fact(n-1)

Why does Stack Overflow error occur in recursion?

If the base case is not reached or not defined, then the stack overflow problem may arise. In the following code, Fact(5) or Fact(1) will cause stack overflow.

1. **Function** Fact(n):
2.   **if** n==10 **then**
3.   **return** 10
4.   **else**
5.   **return** n*Fact(n-1)

Practice problems:

Common problems

1. Write a recursive implementation of the factorial function. Recall that $n! = 1 \times 2 \times \ldots \times n$, with the special case that $0! = 1$.

2. Write a recursive program to calculate the power of x (x^y), where y is a non-negative integer.

3. Write a recursive program to print the nth Fibonacci number.

4. Write a recursive program to check if a given string is a palindrome or not (not case sensitive, ignore whitespaces)

Sample input	Sample output
Evil olive	True
Too bad	False

Numbers

5. Write a recursive program to print the even numbers in a given range.

Sample input	Sample output
3 10	4 6 8 10

1D array:

6. Write a recursive program to print an array of size n in given order.

7. Write a recursive program to print an array of size n in reverse order.

8. Write a recursive program to find the sum of the elements of an array of size n.

9. Write a recursive program to find the products of the elements of an array of size n.

10. Write a recursive program to find the maximum of the elements of an array of size n.

11. Write a recursive program to find the minimum of the elements of an array of size n

12. Write a recursive program to find the average of the elements of an array of size n

13. Write a recursive program to print the odd/even numbers of an array of n integers

14. Write a recursive program to print the prime numbers of an array of n integers

15. Write a recursive program to count the odd/even numbers of an array of n integers

16. Write a recursive program to count the prime numbers of an array of n integers

2D array

17. Write a recursive program to find the maximum of a 2d array.
18. Write a recursive program to count the prime numbers of a given 2d array.

Series

19. Find the sum of the following series up to n^{th} position / Print the following series up to n^{th} position.

    ○ $1 + 2 + 3 + ...$

    ○ $1^2 + 2^2 + 3^2 + ...$

    ○ $1 * 3 + 2 * 5 + 3 * 7 + 4 * 9 + ...$

    ○ $2 * 3 + 4 * 5 + 8 * 7 + 16 * 9 +...$

    ○ $2 * 3 * 4 + 4 * 5 * 3 + 8 * 7 * 2 + 16 * 9 * 1 +...$

GCD/LCM

20. Write a recursive program to find the GCD of x and y where x, y are positive integers. (Hint: use Euclid's algorithm. Two ways to solve this.)
21. Write a recursive program to find the LCM of x and y where x, y are positive integers. (Two ways to solve this)

Digits

22. Write a recursive program to count the number of digits of an integer.

23. Write a recursive program to find the sum of digits of an integer.

24. Write a recursive program to check if a given positive integer is a palindrome or not. An integer is a palindrome when it reads the same backward as forward.
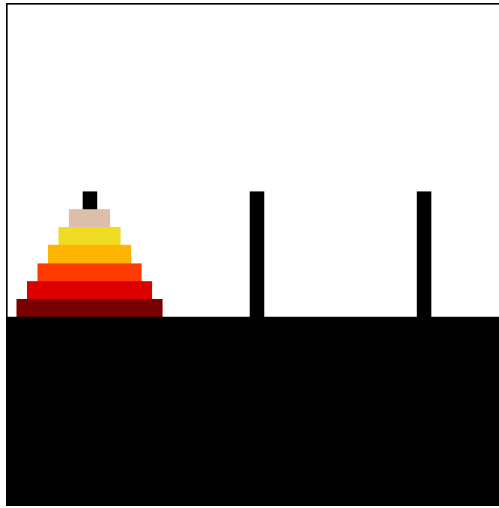
    ○ Try solving it here.

Subset

25. Write a recursive program to print all subsets of a set of n elements.

26. Write a recursive program to print all subsequences of a string.

Miscellaneous:

27. Write a recursive implementation of binary search in a sorted array.

28. Given a set of parentheses check if they are balanced or not using a recursive function.

29. Implement DFS using recursion to traverse a graph.

30. Implement in-order, preorder and postorder traversal of a graph using recursion.

31. Print the path from a node to the root of a binary tree using recursion.

32. Write a recursive program to solve the Tower of Hanoi problem for n disks.

Sample input	Sample output
4	Move disk 1 from A to B Move disk 2 from A to C Move disk 1 from B to C Move disk 3 from A to B Move disk 1 from C to A Move disk 2 from C to B Move disk 1 from A to B Move disk 4 from A to C Move disk 1 from B to C Move disk 2 from B to A Move disk 1 from C to A Move disk 3 from B to C Move disk 1 from A to B Move disk 2 from A to C Move disk 1 from B to C