Multiple Pecursion -> process in which a function makes > 2 recursive calls -> common example is enumerating various configurations in order to solve a combinatorial puzzle if the # of possible configurations is not too large, we can use an algorithm to consider all possibilities syskmatically. In general, the algorithm commerates and tests all k-length sequences w/o repetitions of ekments in given universe U. Build requenus by: 1.) Perursively generating sequences of k-1 elements 2.) Appending each such sequence an element not already in it Puzzle Solm (K, S, U); for each e in U Add e b end of 5 Remove e from U if k == 1: Test whether 5 is a config. Which solves puzzle if 5 solves puzzle then return "Solution" + S PuzzleSolne(x-1, S, U) Permove e from end of S Add e back to U Recursion True PuzzleSolvel3, (1, Ea, b, c) Puzzlesolve(2,4, 25,63) Puzzle Solve (2,6, 20,03) Puzzle Solve (2, c, {a, b}) Puzzle Solve (1, en, 25}) Puzzle Solvell, ab, Ec PuzzleSolveli, ha, Ec3) Puzzle Solve (1, ch, [a]) PuzzleSolve (1, bc, 243) Puzzlesolve (1, ac, {b})

-

3

-3

-3

-3

. .