

## Recursive Strategy: Print a list of integers BACKWARDS

<b>Problem</b>	Print a list of integers in reverse order, e.g. [ 3, 14, 28, 7] Would print 7, 28, 14, 3
<b>What is the smallest version of this problem?</b> <i>(leads to base case)</i>	If the array consists of one element, print the only element (which happens to be the 0 <sup>th</sup> element)
<b>What recursion strategy should I use:</b> <ul style="list-style-type: none"> <li>• <b>Forwards recursion</b> <i>(each recursive step gets larger; the base case is based on # of iterations)</i></li> <li>• <b>Backwards recursion</b> ← classical <i>(each recursive step gets smaller until the base case is reached)</i></li> </ul>	Backwards recursion
<b>For recursive cases, should I:</b> <ul style="list-style-type: none"> <li>• <b>Process first and recur last?</b> <i>(process as I move <u>up</u> the recursive stack)</i></li> <li>• <b>Recur first and process last?</b> <i>(process as I move <u>down</u> the recursive stack)</i></li> </ul>	Process as I move <b>down</b> the stack.  Note that the first time we print will be the base case of [ 7 ]. The second time we print will be the next-to-last case of [ 28, 7 ]. And so on...
<b>What should each recursive step do?</b>	Print the <u>first</u> integer of the Array
<b>For backwards recursion solutions:</b> <b>How should the problem be reduced on each step?</b>	Send progressively smaller Arrays on each iteration where each Array is missing its first element.*
<b>For forwards recursion solutions:</b> <b>How should I keep track of the running answer?</b>	N/A

\*Instead of sending smaller and smaller arrays each time, could you also send the **same array** to each step as well as the size of the smaller virtual array (which would decrement by one on each recursion)?

(The “virtual array” would consist of the last x elements, where x = the progressively reducing size.)