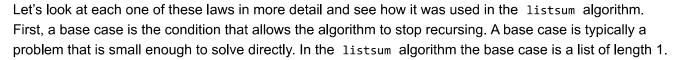
5.4. The Three Laws of Recursion

Like the robots of Asimov, all recursive algorithms must obey three important laws:

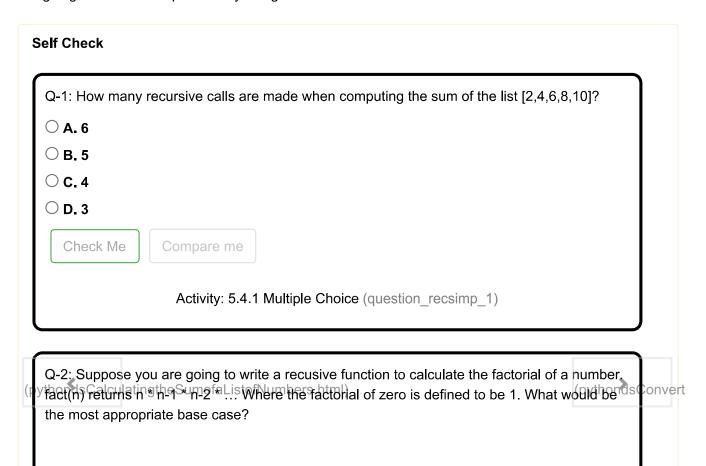
- 1. A recursive algorithm must have a base case.
- 2. A recursive algorithm must change its state and move toward the base case.
- 3. A recursive algorithm must call itself, recursively.



To obey the second law, we must arrange for a change of state that moves the algorithm toward the base case. A change of state means that some data that the algorithm is using is modified. Usually the data that represents our problem gets smaller in some way. In the <code>listsum</code> algorithm our primary data structure is a list, so we must focus our state-changing efforts on the list. Since the base case is a list of length 1, a natural progression toward the base case is to shorten the list. This is exactly what happens on line 5 of ActiveCode 2 (pythondsCalculatingtheSumofaListofNumbers.html#lst-recsum) when we call <code>listsum</code> with a shorter list.

The final law is that the algorithm must call itself. This is the very definition of recursion. Recursion is a confusing concept to many beginning programmers. As a novice programmer, you have learned that functions are good because you can take a large problem and break it up into smaller problems. The smaller problems can be solved by writing a function to solve each problem. When we talk about recursion it may seem that we are talking ourselves in circles. We have a problem to solve with a function, but that function solves the problem by calling itself! But the logic is not circular at all; the logic of recursion is an elegant expression of solving a problem by breaking it down into a smaller and easier problems.

In the remainder of this chapter we will look at more examples of recursion. In each case we will focus on designing a solution to a problem by using the three laws of recursion.









○ A. n == 0
○ B. n == 1
○ C. n >= 0
○ D. n <= 1
Check Me Compare me
Activity: 5.4.2 Multiple Choice (question_recsimp_2)

You have attempted 1 of 3 activities on this page

user not logged in

(pythondsCalculatingtheSumofaListofNumbers.html)

(pythondsConvert