1. State the definition of recursion.

A recursive algorithm is one that relies on the results of its own computation.

2. True or false:

a. *All* algorithms can be stated recursively. False

b. Some algorithms can *only* be stated recursively. False

c. For most people, recursive algorithms are more difficult to discover and understand than nonrecursive algorithms. True

d. All recursive algorithms must have an escape clause. True

e. Recursive algorithms, if not properly coded, can terminate with a stack overflow error; if true, why? True. If there is no base case then the program will just keep calling itself. Every call adds all the current variables to the stack. Eventually, it will overflow. A special kind of infinite loop.

3. The menu of Juan's Taco Tower states “when ordering a Sombrero Meal you get a taco, and a soda, and your choice of an ice cream sundae *or* a Sombrero Meal.”

a. What part of the Sombrero Meal's description is recursive?

The "or" clause at the end. There are two choices: Meal = (Taco&&soda&&icecream)||Meal or Meal=(taco&&soda)&&(icecream ||Meal).

The comma after soda means that the following clause is an independent clause, this means that the second parsing is the correct one.

Note: The answers in part c assume that the or is an inclusive or. If it is meant to be an exclusive or then number c2,c3 and c5 would be no. All the others are still yes.

b. What is the base case (escape clause) in the description of a Sombrero Meal?

(Taco&&soda)

c. Can I order a Sombrero Meal consisting of:

1. 1 taco, 1 soda, and 1 ice cream sundae? Yes.

2. 2 tacos, 1 soda, and 1 ice cream sundae? Yes

3. 2 tacos, 3 sodas, and 1 ice cream sundae? Yes

4. 1 taco and 1 soda? Yes (no ice cream will be given). At worst, the customer throws out the sundae.

5. 1 taco, 1 soda, and 2 ice cream sundaes? Yes. ("or" is true if both are true)

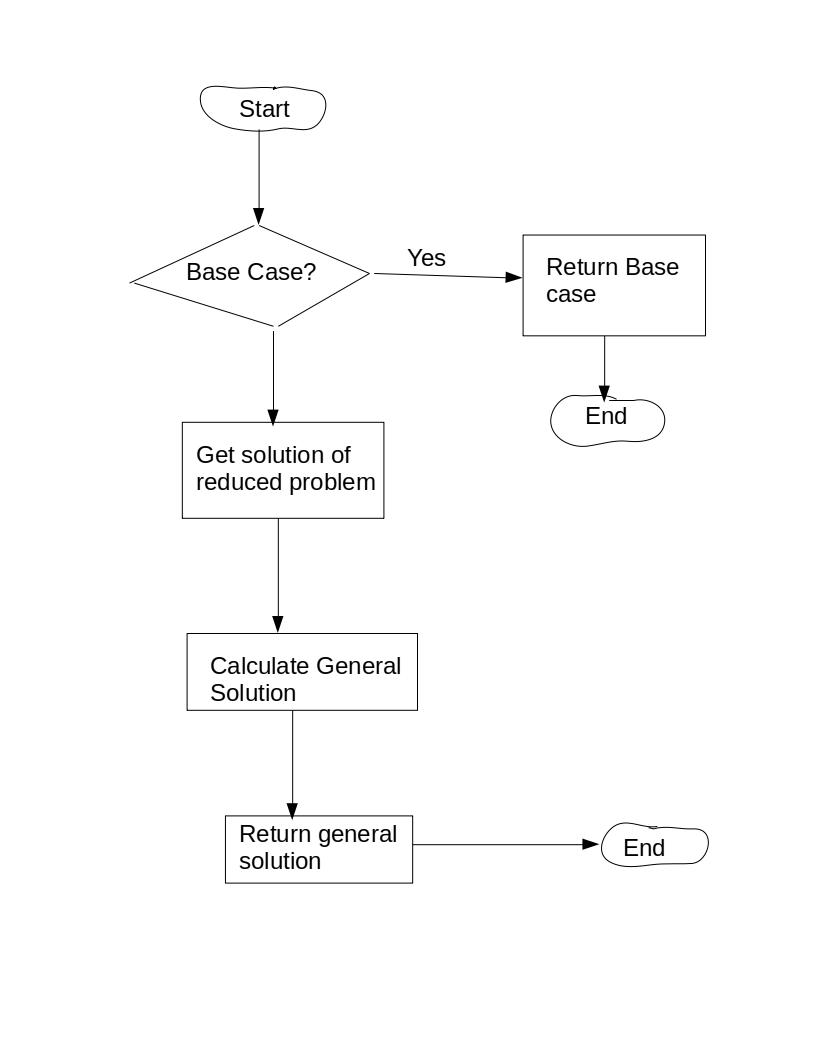
4. Define:

a. Base case - An example of the overall problem that is easy to solve. eg. 0! = 1.

b. Reduced problem - This is tricky to define. It is a problem in between the original problem and the base case that reduces to the base case. It basically is the heart of the recursion. At one end is the base case and at the other is the original problem. I think of it as a link in a chain.

c. General solution - The general solution tells us how to go from one link of the chain to the next.

5. Give the generic flowchart for a recursive method that has one base case. Note: You may draw your flowchart using some another application, but copy and paste the image of it here.



6. Give the four steps in the methodized approach to formulating a recursive algorithm.

1) Determine the base case.

2) Determine the reduced problem.

3) Determine the general solution.

4) Combine 1,2,3 to code the algorithm.

7. Give an advantage of a recursive algorithm over an iterative algorithm.

I think they are more intuitive and easier to code.

8. Give two advantages of an iterative algorithm over a recursive algorithm.

The iterative approach is generally faster and uses less memory than recursion. Recursive algorithms can also be harder to debug.

9. A recursive method is used to calculate 64. The base case is 60.

a. What is the first value returned from the recursive invocations?

1

b. What is the second value returned from the recursive invocations?

6