Recursion

3/31/2009
Opening Discussion

- Do you have any questions about the quiz?
- Let's look at solutions to the interclass problem.
- Do you have any questions about the assignment?
- Do you have any questions about the reading?
You should have learned about recursive functions in PAD1. A recursive function is simply a function that calls itself.

You can use recursion to imitate loops, but we won't do that very often in C or Java. Where recursion comes in really handy is when a function needs to test more than one alternative at a time.

This works nicely because the call stack remembers where you are in a given function so when you return back, you can take off from that point again.
One of my favorite recursive algorithms is maze solving. This is a special case of graph traversals which are common problems in CS.

We'll use a 2D array of ints as our maze and we can even put this into our drawing program.

A simple warm-up piece of code is flood fill like you would have in a drawing program.

Once we have that we can see how to convert it to do things like find the shortest path through a maze or count all paths through a maze.

We can try to make this nice and graphical as well so it fits properly into our drawing program.
Another one of my favorite recursive algorithms is formula parsing. This allows us to have the user type in a function and our code can evaluate it.

We do this through “divide and conquer”. We split the formula in two across the lowest precedence operator then recursively evaluate the two halves.

We can use this to put function plotting or animation into our program if we give it the ability to handle a variable.
- Do you have any questions about recursion? Do you see how it can do things like search a space of possibilities or divide up a problem?
- The design for assignment #5 is due Thursday.
- Interclass Problem - Write a recursive function to find the longest path through a maze.