Recursion: Part 3

10-27-2003

Opening Discussion

- What did we talk about last class?
- Do you have any questions about assignment #6?
- Practice with recursion.

Debuggers

- You can get help with finding runtime errors, and to some extent logic errors as well. Hopefully you have already used the practice of putting extra printf statements in code to help you figure out what it is doing. If you compile with the -g option you can also use a debugger.
- The debugger on these systems in gdb. You run "gdb a.out" (or whatever your program name is). Typing run at the prompt starts your program. You can pipe things in there too.

More on the Debugger

- When a fault occurs (could be you hitting ctrl-C) you can type in "where" to get a stack trace including line numbers.
- The command print will print out variable values.
- Use quit to get out.
- There are also many other powerful features in the debugger that allow you to set break points, return from functions, continue after breaks, etc. Use help to see these.

Recursion in Multiple "Directions"

- Last time we looked at recursive functions that call themselves more than once.
- These functions can do things that aren't easily done with loops.
- What really happens is that it completes one of the calls completely, then makes the others. The memory of the stack is what allows it to come back and go in other directions.

Divide and Conquer

- Another example of where recursion can be very useful is in algorithms that are called divide and conquer algorithms.
- In these algorithms, the problem is repeatedly broken into smaller pieces until it gets to something that can be easily dealt with.
- It then combines the small pieces to get the final answer.

Simple Examples

- Let's write code for doing some simple problems through divide and conquer. These problems display the technique, not so much their power.
  - Sum the elements of an array.
  - Find the minimum element in an array.

Maze Example

- Along the lines of the flood fill are maze problems, let's write a function that tells us how many ways there are to get from a start point to a finish point in a given maze.

NULL Pointers

- In our earlier discussion of pointers we looked at how to declare pointer variables and assign them the address of other variables.
- Sometimes we also want a value that we can use when a pointer doesn't point to a valid address instead of just garbage.
- NULL is this value. It is defined in stdlib.h.

Minute Essay

- Write code that uses divide and conquer to count the number of even numbers in an array of integers.
- I won't be here Wednesday and you have the day off. On Friday we'll start back up with files. Remember that you should be working on assignment #6 which is due a week from today.