Slide 1

Administrivia

- Reminder: Quiz 2 Monday. Likely topics conditional execution, functions. (But no recursion.)

Slide 2

Minute Essay From Last Lecture

- Many people said they'd seen recursion, but I wonder if what some mean is what I'd call "repetition" (a more general term), since one person mentioned loops (which are our next topic).
- Many people who lost points on the quiz said they had just made silly mistakes. I hope the next one goes better!

Slide 3

Recursive Functions — Recap/Review

- A recursive function is one that calls itself (except for its "base case(s)").
- An obvious(?) use of recursive functions is to compute mathematical functions defined recursively. But there are other uses! The ones we'll do now are somewhat contrived, but the concept turns out to be very powerful for some more-advanced purposes.

Slide 4

Recursive Functions — Basic Idea

- Identify one or more "base cases" — inputs we can handle directly with no recursion.
- For other cases, figure out how to get the result based on one or more "smaller" instances of the same problem.
- Sometimes it helps a lot to be very clear about what the recursive function does. This is fairly straightforward for recursive functions based on recursive math definitions, but for other things . . .

Slide 5

Recursive Functions — Sum Example

- In sum example from last time, the recursive function's job is to sum up all (remaining) input and return that sum.
- First step in recursive function is to try to read an `int`.
- If there's not one, that's the base case — no more input, sum is 0.
- If there is one, sum is that input, plus the sum of all remaining input — which we find by making a recursive call.

Slide 6

Recursive Functions — Countdown Example

- Now try writing a recursive function to count down from N to 0.

Slide 7

A Little About Character Data

- (We can't work with strings of character data until we know about arrays. But we can work with single characters, which allows for more kinds of examples.)
- Single characters represented by type `char`. 7-bit, usually ASCII, so range is enough to represent digits, alphabetic characters (upper- and lower-case), various punctuation. Not enough for all non-English languages, alas.
- Worth noting that `char` values are a subset of `int` values, so functions for working with characters sometimes take/return `int`s.
- Can do I/O with `scanf` and `printf`, but simpler to use `getchar` and `putchar`.

Slide 8

Recursive Functions — Another Example

- A trivial but fun(?) example of using recursion: Print line of text in reverse order.

Slide 9

Minute Essay

- Any questions?