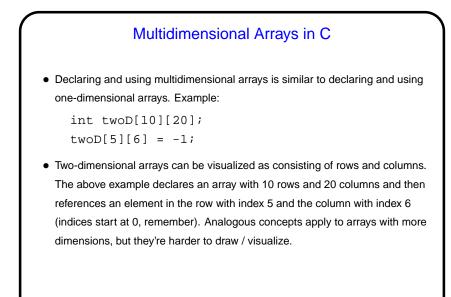


Slide 2



Slide 3

## Multidimensional Arrays in C, Continued With the older standard for C (C89), passing multidimensional arrays to functions is ugly, unless the size of the array is known at compile time. The variable-length arrays of the newer standard seem to allow things to be done more nicely.

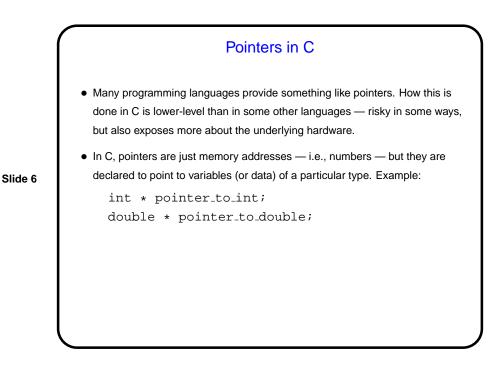
Slide 4

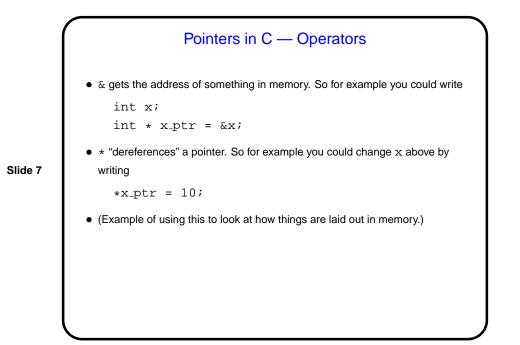
## **Pointers**

• Every time you call scanf, you pass it at least one parameter of the form &x. What does that mean? Also, when you look at man pages for some functions, they show function declarations with parameters of the form *type* \*. What does that mean?

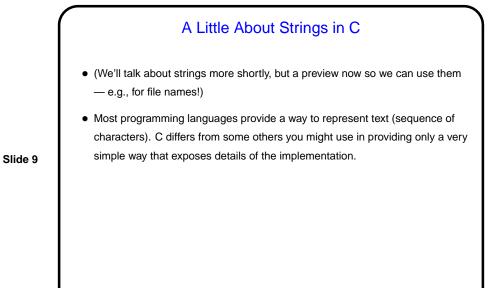
Slide 5

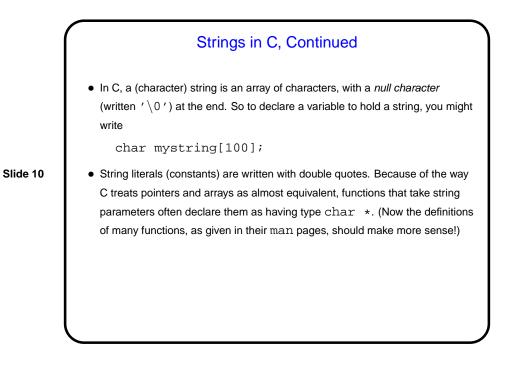
To explain, we need one more kind of variable — *pointers*. A pointer, as its name suggests, points to something — namely, a location in memory.
 Typically a pointer "points to" a variable.

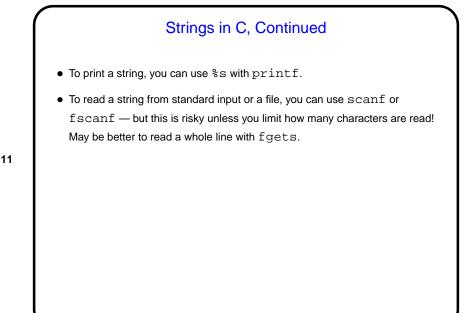




Pass By Reference
A significant limitation we've had to deal with is that functions can only return a single value. Pointers provide a way to get around this restriction: By passing a pointer to something, rather than the thing itself, we can in effect have a function return multiple things.
Slide 8
To make this work, typically you declare the function's parameters as pointers, and pass addresses of variables rather than variables. (This is how scanf does what it does, and why you need the &.)
(Example.)







Slide 11

