Slide 1

Administrivia

- (None?)

Slide 2

Arrays and Functions

- As noted previously, you can operate on individual elements of an array as if they were single variables (use them in expressions, assign to them, and pass them to functions); syntax is name of array followed by index in square brackets.
- You can also pass a whole array to a function; syntax on calling side is to just give its name (no index); on function side, follow name with brackets. *Note* that in this case the function actually has access to the array and can change its elements. (Is this an exception to the rule about “pass by value” with copying? Not really — what is being passed is a pointer — but it may appear so.)
- One thing to know is that information about how big the array is has to be provided to the function separately and explicitly. You do this slightly differently for old-style arrays and VLAs. (Example.)

Slide 3

Multi-Dimensional Arrays

- Single-dimensional arrays provide a way to represent something like singly-subscripted variables in math. What about variables with multiple subscripts? e.g., matrices? “multi-dimensional arrays”
- C has them (syntax in book), but they’re somewhat awkward to work with . . .

Slide 4

Multi-Dimensional Arrays, Continued

- For old-style arrays (i.e., not VLAs), can’t really write functions that work with different sizes, because to locate an individual element you need information about (some) dimensions of array (e.g., number of columns for 2D).
- For VLAs, functions are easier but total size may be limited, and some very cautious programmers avoid VLAs because some compilers allegedly do not support them well.
- Dynamic allocation (making an array of arrays — more later) may be better but is tedious.
- User-defined macros that “fake” multiple dimensions in single-dimensional array also work okay but are tedious.

Slide 5

Multi-Dimensional Arrays, Example — ASCII Art

- That said, an example may still be useful:
- We could write a simple “ASCII art” program that “draws” pictures using characters only, with:
  - a two-dimensional array of `char` as the “canvas”, and
  - a simple text-menu-driven interface to print, set blocks, clear.

Slide 6

Minute Essay

- Can you think of problems you might want to solve that would require multi-dimensional arrays?