

Slide 1

### Administrivia

- Reminder: Homework 6 due today.
- Homework 7 on the Web. Due in a week. This one is also not easy but I hope is at least a little interesting?

Slide 2

### Minute Essay From Last Lecture

- Most people had seen arrays in Matlab. So the concept is maybe not new, just the details?
- About uses for arrays, there were mentions of matrices and systems of equations. Both good uses of multidimensional arrays!

### Arrays — Review/Recap

Slide 3

- Arrays give you a way to do something akin to subscripted variables in math: You reserve space for a group of values of a particular type, giving a name, and then reference particular values with that name and an *index* (like a math subscript).
- Valid indices range from 0 to one less than the array size. Alas, in C it's all too easy to use an index not in that range, and results are — unpredictable. (Your program might crash, or it might overwrite some other variable.) (Worth noting that more-modern programming languages have safety checks to prevent this. C doesn't do that, for reasons its adherents think good.) Fix example from last time.

### Arrays — More Examples

Slide 4

- As one more example, we could write the program mentioned last time, the one that counts how many of each character in input.

## Arrays and Functions

Slide 5

- 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 — though 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 next time.)

## Multi-Dimensional Arrays

Slide 6

- 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 . . .

### Multi-Dimensional Arrays, Continued

Slide 7

- 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.
- (Example soon.)

### Minute Essay

Slide 8

- None really — just sign in, unless questions?