Administrivia

- Reminder: Homework 7 due today.

- Reminder: Quiz 4 Monday. Likely topic is arrays.

- Homework 8 on the Web. Due next Friday. Only one problem, but another not-easy one.

Slide 1

Minute Essay From Last Lecture

- Most people said they were very busy with other courses. Fair enough. It happens?

Slide 2

Text Data — Single Characters

Slide 3

- `char` is considered an integer type and can be worked with as such. Note that while these days ASCII is by far the most common encoding, standard doesn't require that, and there are other possibilities.

- Many library functions for working with single characters (e.g., `isalpha`).

- Character literals represented using single quotes.

- Can read in / print single characters with `scanf` or `printf` using `%c`. Or can use `getchar`, `putchar`. Note that `getchar` returns an `int`. Why? so it can return special value EOF when no more input.

Text Data — Strings

Slide 4

- Most more-recent languages have nice ways of working with "strings" of text data that hide details and provide nice functionality.

- C, in contrast, provides a bare-bones version, in which text strings are represented as arrays of `char`, with an end-of-string character (`'\0'`) that allows an array of fixed size to store strings of different sizes. Simple but subject to all the perils of arrays!

- String literals represented using double quotes. Can include "escape" characters (e.g., `'\n'`.)

Text Strings — Output

- Can use `printf` with `%s`.

- Can also use `puts` (which adds a newline).

- (Similarly for `fprintf` and `fputs`.)

Slide 5

Text Strings — Input

- Surprisingly (or not, given C's minimalist implementation of arrays), no nice way to do this!

- Can use `scanf`, but no nice/general way to be sure you don't overflow array, and getting something that includes whitespace may be tricky.

Slide 6

- Can get a whole line with `fgets`, but must provide a fixed-size array (so, what size?) and deal with newlines.

- `gets` looks useful but observe what its `man` page says(!).

- Consider processing data character by character, or using command-line arguments.

Working With Text Strings in C

- Once you have some "strings" in your program, what can you do with them?

- You can work on them as arrays of character (that's what they are) or using pointers (as in the example earlier with an array of `int`s).

  (Example.)

Slide 7

Minute Essay

- Questions? otherwise just sign in.

Slide 8