Slide 1

Administrivia

- (None?)

Slide 2

Minute Essay From Last Lecture

- Almost everyone got right answers for the question about bit manipulation. (I have expected answers in the last slide of the notes.)

  Many said they'd seen something like this in another course, usually in the context of digital logic or circuit design, but many had not. Interesting?

- No comments about Homework 9 really stood out, though several people did mention that it seemed longer and/or more difficult. A couple did mention that it was interesting or "cool". I hoped it would be!

  Homework 10 is similar in that it initially may look intimidating, but you get a lot of starter code, and my hope is that the parts you have to complete will not be too daunting. And you end up with a program that does something that at least I think is interesting!

Slide 3

Lists and Other "Collection" Types

- We know about arrays as one way to represent a collection of data.
- We could abstract from this a bit and talk about "lists" (what the textbook calls "linear lists") as linear ordered collections of data.
- We could also consider coming up with ways of representing non-linear collections such as trees, graphs (in the sense of a collection of nodes and edges), etc.
- Many/most programming languages support this idea, sometimes through fairly extensive libraries. C, not surprisingly, doesn't, but you can build your own, typically using `struct`s and pointers.

Slide 4

Linear Lists

- One way to implement a linear list is with an array. Simple and efficient if list can be of fixed size and you don't need to add/insert elements in the middle.
- If that doesn't work well, an alternative is a "linked list" consisting of a collection of "nodes", each consisting of a list element plus a pointer to the next element.

Slide 5

Linked Lists in C

- Defining a `struct` for the nodes of a linked list is somewhat tricky in C because one of the fields needed is a pointer to something of the same type. But the following works to define items in a linked list of `int`s:

```
typedef struct int_list_node {
    int data;
    struct int_list_node * next;
} int_list_node_t;
```

- (Start looking at example code for sorted list.)

Slide 6

Minute Essay

- None — quiz.