Slide 1

Administrivia

- Homework 6 on Web; due in a week.
- Quiz 5 will be next Thursday (not Tuesday).
- (Review minute essay from last time.)
- (Review examples from last time.)

Slide 2

Arrays in C — Recap

- Basic idea is to provide something analogous to subscripted variables in math.
- Declarations look like other variable declarations, except name is followed by [n], where n is the number of elements.
- Reference individual elements with variable name followed by equivalent of subscript in math — [i], where i is an index *ranging from* 0 *to* $n - 1$. No checks in C to make sure the index is really in that range!

Slide 3

Arrays in C, Continued

- Declaring that a function parameter is an array — put brackets after parameter, usually don't specify array size as part of array parameter. Example:

  ```
  double array_sum(int size, double nums[]);
  ```

  (The newer C99 standard supports a slightly different syntax. More about this another time.)

- Passing an array to a function — just give its name. Example:

  ```
  double stuff[100];
  printf("%f\n", array_sum(100, stuff));
  ```

Slide 4

Arrays in C, Continued

- Using an array parameter within the function — as in previous examples.
- A difference between array parameters and other parameters — array elements can be changed.
- If your function isn't supposed to change the array, declare the parameter `const`, e.g.

  ```
  double array_sum(int size, const double
  nums[]);
  ```

  (Helps people using your function understand its effects, allows compiler to enforce that no changes are made.)

- (Example(s) . . .)

Slide 5

Minute Essay

- None — sign in.