

Slide 1

### Administrivia

- Homework 2 grades mailed.
- Sample solutions for first two homeworks posted, at the bottom of "lecture topics" etc. page.
- Reminder: Homework 3 due today.
- Homework 4 posted. Two problems, somewhat less simple than previous assignments. Due in two weeks.

You should be able to do the first problem based on this week's video lectures; you might want to wait on the second until after the next group of lectures, though you could start earlier if you like.

Slide 2

### Minute Essay From Last Lecture

- Most people are planning to just use the online tutorial, though a few are buying the textbook.

Slide 3

### Homework 2 Essays, Etc.

- Pretty much everyone got full or nearly-full credit. Yay! A good start.
- Most people seemed to find the assignment a reasonable way to start getting used to C.
- Some people mentioned finding the math a little more complicated than they thought. Others found it straightforward, often because they'd done a similar program in another language.
- Problems mentioned included forgetting semicolons and C's somewhat different approach to producing printable output. (But one person mentioned liking the latter! I do too.)

Slide 4

### Simple I/O, Revisited

- Doing a really good job with interactive input is surprisingly tricky — what constitutes an error, how do you prompt user to try again.
- So for this class we'll focus on some simple safety checks: if input should be numeric it is, and values make sense for the program (e.g., inputs to GCD program are not both 0). I like to always print input values so users can at least confirm that what they thought that typed in is what the program read.
- Some online sources discourage use of `scanf`. There are reasons for getting input other ways, but I say they have their problems too. It *is* annoying that it doesn't detect overflow, but oh well.
- For this class it's usually best to just bail out on bad input, rather than retrying. (And if you do anything else on homework, it breaks my semi-automated testing.)

### Recap of Video Lectures

Slide 5

- Loops in C (`while` and `for`).  
Most people used `while` for the quiz question on first lecture, but several used `for`. “Hm!”?
- Arrays in C.  
Why no checking of array indices . . . Partly an efficiency measure, but also no way to do it in general without storing length with array.
- Questions?

### Practice with Loops and Arrays

Slide 6

- Last week we wrote a program that prompts for a sequence of integers (any number, ending with input that's not an integer) and prints them in reverse order. Revise it to use an array and loops rather than recursion.
- As before, you can work individually or in pairs. When you have something:  

```
chmod go+r pgm.c  
cp -p pgm.c /users/bmassint/TEMP1120/yourusername
```
- At end of class we'll look at solutions.

### Minute Essay

- If your solution to Homework 2 made use of constants such as 3600 for seconds per hour, did you get them from computing them (60 times 60) or from your favorite search engine? (I'm curious!)

Slide 7