

Slide 1

Slide 2

## Homework 6 Essays Not much really stood out ... Some people said the problems initially seemed difficult or time-consuming, but turned out not to be given all the sample code. Some people mentioned that the last problem (encryption with a function using pointers into alphabets) helped them understand pointers better. That was one of the goals! A couple of people mentioned finding encryption interesting, and one mentioned how breaking the German Enigma machine code was significant in the history of computing. Indeed, and interesting!

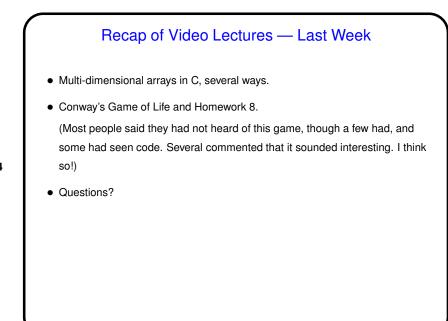
1



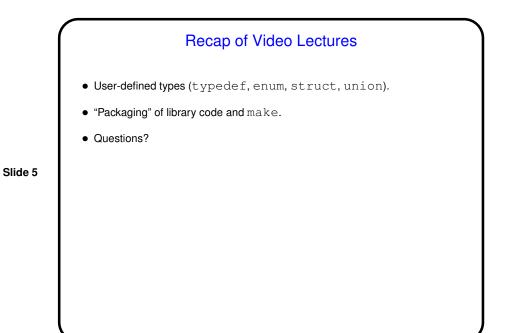
 On the second problem, surprisingly few people remembered that before blithely putting something into an array *you need to be sure the index is valid*. Remember that C does no checking of array indexes, and the results of out-of-bounds access range from crashes to subtle hard-to-find bugs!

Slide 3

 Most people, however, did do the third problem with the recommended approach. (In previous semesters this was often not true!)



Slide 4



Homework 8 and make
Two programs, with a lot of common code. So I've structured the starter code with the likely common code separate from the main program(s).
Note setup: Common code split into .h (declarations) and .c (definitions). *Very* typical packaging for library code. Makefile says how to build.

Slide 6

## 3

