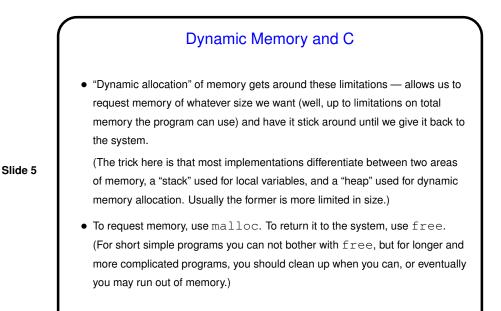


Dynamic Memory and C With the old C standard, you had to decide when you compiled the program how big to make things, particularly arrays — a significant limitation. Variable-length arrays help with that, but don't solve all related problems: In most implementations, space is obtained for them on "the stack", an area of memory that's limited in size. You can return a pointer from a function, *but* not to one of the function's local variables (because these local variables cease to exist when you return from the function).

Slide 3



Dynamic Memory and C, Continued
Examples:

int * nums = malloc(sizeof(int) * 100);
char * some_text = malloc(sizeof(char) * 20);
free(nums);

Book recommends "casting" value returned by malloc. Other references recommend the opposite! But you should check the value — if NULL, system was not able to get that much memory.
(Redo sort/search example using dynamically allocated memory.)

