

Slide 1

Administrivia

- Homework 4 on Web; due next Monday.

Slide 2

Dynamic Memory and C

- With the C89 standard, you had to decide when you compiled the program how big to make things, particularly arrays — a significant limitation.
- Variable-length arrays in C99 standard 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).

Dynamic Memory and C, Continued

Slide 3

- “Dynamic allocation” of memory gets around these limitations — allows us to request memory of whatever size we want (well, up to limitations on total memory the program can use) and have it stick around until we give it back to the system.

(The trick here is that most implementations differentiate between two areas of memory, a “stack” used for local variables, and a “heap” used for dynamic memory allocation. Usually the former is more limited in size.)

- To request memory, use `malloc`. To return it to the system, use `free`.
(For short simple programs you can not bother with `free`, but for longer and more complicated programs, you should clean up when you can, or eventually you may run out of memory.)
- Compare/contrast with Java — allocate space for objects with `new`, no explicit deallocation, garbage collection.

Dynamic Memory and C, Continued

Slide 4

- 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.

Example — Revised Sort Program

- First change the program to allow specifying at runtime that N inputs are to be generated.
- Next change it to accept input from a file. (Not done in class.)

Slide 5

Minute Essay

- None — sign in.

Slide 6