Outline for May 31, 2002

Handouts:*The Dynamic Debbugger gdb*
Reading: Johnsonbaugh and Kalin, pp. 679–702 (appendix of useful functions)

1. Greetings and felicitations!
2. Miscellaneous
   a. terminate program (exit); include <stdlib.h>
   b. sort array of data (qsort); include <stdlib.h>
   c. time of day (time, ctime); include <time.h>
   d. execute cvommand (system); include <stdlib.h>
3. Debugging
   a. programs have bugs; find and fix them
   b. static debugging: insert debugging code into source, recompile and run
   c. dynamic debugging: look at the program as it runs, observing (and maybe changing) variables, etc.
4. Static debugging
   a. using printf to print variable values; mention %p (prints pointer value, usually as a hex integer)
   b. using printf to print where you are (ie, on function entry printf("in function\n");
   c. #ifdef DEBUG … #endif around the printfs so you can leave them in the source if you need them again
   d. assert(x) macro: assert(0 <= i && i <= n) causes program to exit with error message if (0 <= I && I <= n) is false; must include <assert.h>. To delete, say #define NDEBUG and they will not be in the compiled code.
5. Dynamic debugging
   a. debugging tool instruments executable program so it can be stopped, examined, altered, and continued interactively
   b. go through the handout
   c. mention the "where" command which shows you the program stack