Dynamic Memory Allocation

- How do you allocate an array of variable length?
- Example: reading in different data sets of different lengths
- Answer: Allocate as a pointer, then later allocate the necessary memory for the length of the array
- Use function `malloc` in stdlib to allocate
- or `calloc` to allocate and initialize to zero
- `malloc` has one argument — the amount of memory to allocate
- `calloc` has two arguments, the number of items and the amount of memory each one takes

`malloc.c`

Command Line Arguments

- Can pass values to the program when you run it
- Example: the name of the datafile and its length
- `main` takes two arguments: `argc` and `argv`
- `argc` is the count of the number of arguments in the unix call of the program (including the name of the executable)
- `argv` is an array of character strings, the values of the arguments, with `argv[0]` being the name of the executable
- If you want to use an argument as a number, you need to convert it from a string to an int or float/double with `atoi` or `atof` from stdlib
- The `main` call in the program must specify its arguments, e.g., `int main(int argc, char *argv[])`

The `qsort()` function

- To sort in C, use the `qsort()` function from stdlib.
- `qsort()` takes four arguments:
  - a pointer to the array to be sorted
  - the length of the array
  - the size of an element of the array
  - a comparison function, `fcmp(a,b)`, that returns -1 if a<b, 0 if a=b, 1 if a>b
- It is complicated so that the same function can be used for different types of variables

`sortvec.c`

Allocating Variable Matrices

- This is easiest with the *Numerical Recipes* code
- See `setup.c` and `setup.h`
- A matrix must be treated as a pointer to pointers

`boxing.2.c`

Pseudo-Dynamic Memory Allocation with GCC

- gcc can do run-time memory allocation
- Not truly dynamic, so "pseudo-dynamic"
- gcc will let you declare arrays and matrices in terms of previously defined variables
- This is **NOT** ANSI standard

sortvec2.c

Declaring local variables

- Variables can be declared in a local environment by declaring them inside of curly braces `{}`
- In C, you can declare new variables at the beginning of such an environment
- You can combine this with gcc's pseudo-dynamic memory allocation to approximate dynamic memory allocation

`sortvec3.c`