Objectives for 11/8/2017 (Mon)

- **Linked lists: freeing**
  - iterative – see `free_list_iterative.c`
  - recursive – see `free_list_recursive.c`

- **qsort(...) with compare_fn**
  - Relation to HW12:
    - `pq_enqueue(...)` takes a generic `compare_fn`
    - `destroy_node(...)` takes a generic destroy function which follows similar principles

*Note: The `free_list_recursive.c` example was posted after class.*
sort() - sort array of any type

Point objects (with .x and .y fields)

\[ \text{Size (each member)} = 8 \]

\[ \text{h members} = 3 \]

\[ \begin{align*}
  &.x = 5, \quad .y = 6^3 \\
  &.x = 6, \quad .y = 5^3
\end{align*} \]

compare-fn takes address of lhs and rhs for any comparison.
size 0: \quad \text{head} \equiv \text{head} = \text{NULL} \\
\text{free (NULL);}

size 1: \\
\text{head} \equiv \begin{array}{c}
4 \quad \equiv \\
2 \quad \equiv \\
1 \\
\end{array} \\
\text{free (head);} \\

size 2: \quad \text{// free head until empty} \\
\text{// free sublist} \\
\text{head} \rightarrow \text{next} \\
\text{// free head} \\

// may be NON-RECURSIVE (ITERATIVE) RECURSIVE

base case: size 0 or 1