Let's design Lab 3 by Comment

We just create function names as we go and then we expand each function later.

```
/* main: time how long it takes to DoEverything */

/* DoEverything: if we init, read&run the sim and then do any teardown
*/

/* read&run:  if readloop puts balls on the table, run the table */

/* run the table: As long as we have stuff to do, output the world and
change the world, then do final output and free all the dynamic memory
*/

/* stuff to do: we have stuff to do unless the in-play list is empty
*/

/* OutputWorld: (this is a two-liner you've seen before) Output the
world each way that is enabled */

/* OutputText: Do the time and score header, sort on Y, do the in-play
list */

/* OutputGraphics: clear, time, score, in-play list, refresh, sleep */

/* final output: another two-liner, one for each mode */

/* final text: different header, off_table list, newline */

/* final graphics: loop that calls master graphics until 4 seconds go
by */

/* change the world: clock tick, update things, dead balls */

/* read loop: scanf to a static struct, get it launched */

/* get it launched: ask for dynamic memory, copy, convert, try to
insert and deal with failure to insert */

/* update things: move things, constrain things */

/* move things: iterate the in-play list, moving each ball */

/* dead balls: delete balls that are off table from the in-play list,
placing them instead on the off-table list.  Be sure to score it and
give the right message. */

/* free all the memory: delete everything from the in-play list, and
the off table list.  Print how many we deleted from each as we do
```

```
them.  (The in-play list should have been empty. The numbers should
all match.  */
```

Details Left Out

Each “do the X list” means an iterate call for that list with an appropriate action function. There is a knee-deep pile of callback functions to write to support list operations. For example, 2 different one-liner functions; one that handles printing balls on the table and the other handles printing balls that are off table. You also need functions that compare in Y and functions that compare in VY.