

For Safety, Check for End of List in Loop Body

```

player_t** find;
for (find = &player_list;
    p != *find;
    find = &(*find)->next) {
    if (NULL == *find) {
        return 0;
    }
}

```

If we reach the end of the list, `p` is not in the list, so fail.

Remove the Player, Free the Blocks, and Return Success

```

*find = p->next;
free (p->name);
free (p);
return 1;
}

```

Remember that `find` points to the pointer to be changed.

Free the name, then the player.

Return success.

Examine How `player_delete` Works in Detail

Let's do a **detailed example**

- of `player_delete` execution
- on a linked list of three players
- with variables shown in LC-3 memory.

Let's first identify where each variable resides:

- in the global data area,
- in the heap, or
- in the stack.

Dynamically Allocated Data Reside in the ...

The linked list is shown below (head on left).

Where are these data
(global data, heap, or stack)?

