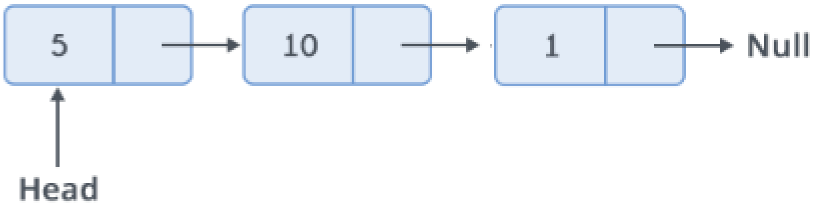
7.5 - The Last Part of Lab5

1. Refer to the full directions for Lab5 if you need to: https://docs.google.com/document/d/1Yh8zNchxV2V3-8jg4dg6PvJlymOa4WivgNeD0dAhdvk/edit?usp=sharing

2. Add the `delete()` function to your `linked_list.c`
   `delete()` should take the head and a value as arguments, traverse the list, and remove the first node in the list that contains the specified value and `free` the memory for that node. For example, if we called:

   `head = delete(head,20)` was called on the linked list that contained 5-10-20-1 the linked list after `delete()` returned would contain:



   *Hints*:
   - You can look at the binary search tree code we wrote in class or a different version of binary search trees in Section 6.5 in Kernighan and Ritchie, although it can be good practice to try to finish the lab without looking!
   - You are essentially splicing out a node from the list. Looking at the above diagram with 5-10-1, if we wanted to remove 10, we would have to:
     - navigate to 5 (one *before* the one we want to delete)
     - be able to get what 10 is pointing to (the `next` of the one we want to delete)
     - set 5`->next` to be 10`->next`
     - `free` 10
   - Also, ***never*** attempt to access a pointer after it has been freed! In the above example, we must deal with 10`->next` before freeing 10.

Turn in your code to moodle in `lab5.5.tgz`
It should have:
- `linked_list.h`
- `linked_list.c`
- `lab5.txt` (that has your name as the driver)

There is a test `main()` for you in:
`/home/msarris/csci235/lab5/test7.5.c`