

In-Section Exercises - Week 6

CS50 — Fall 2011

Prepared by: Doug Lloyd '09

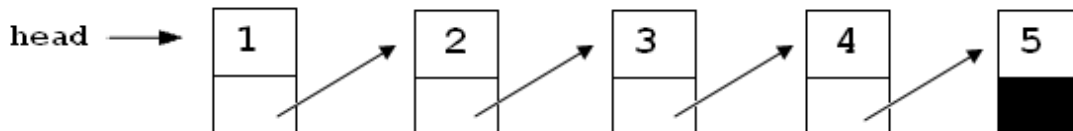
October 17, 2011

1. Define a structure for a problem set, whereby each structure contains:

- A 25-character (maximum) string to hold the author's name
- An integer that represents the problem set's number
- Assuming the problem set contains only one file, a pointer to the file containing the code for the problem set
- A single character representing that person's grade on the problem set

Define this structure both with and without the use of `typedef`, and show how one would declare an instance of each of these structures both with and without the use of `typedef`.

2. Instantiate a structure of that type, with the variable name `<username>_pset5`, where `<username>` is your own user name, filling in the appropriate values for your name and the problem set number. Assume that the code for the problem set is located in `<username>5.c` (which I will want to read, but not edit), and that you got an A on the problem set.
3. Using `enum`, define a type that we can use to represent the twelve months of the year, and make it so that JAN has the value 1, with each subsequent month increasing by 1.
4. Earlier we discussed singly-linked lists and I told you to write `l1ist_delete_node()` as an at-home exercise. Well, I lied. Assume you have the below linked list:



And now we have the following line of code:

```
head = l1ist_delete_node(head, 4);
```

Write the code that will delete that node from the list and will reconfigure the list so the chain is reconnected after you delete the node. Be sure to also account for the possibilities that I wanted to delete the node containing 1 or 5, as well! That is, your function should be written with an eye for generality, not this specific case. Recall that `l1ist_delete_node()` has the function declaration specified below. Your function may not alter this declaration and must return a pointer to the head of the new linked list. Assume, as well, that all of the nodes in the list were dynamically declared (as with `malloc()`), and therefore the memory for them must be released back to the operating system.

```
sllist *l1ist_delete_node(sllist *head, int val);
```