

1 25 Points

Suppose we have a large, complicated program which stores information in a doubly-linked list whose nodes have the following form:

```
struct node {
    int province_code;
    int beheadment_count;
    int rioting_index;
    struct node *next; /* Pointer to next node in list */
    struct node *prev; /* Pointer to previous node in list */
};
```

The first node in the list has a NULL value stored in its `prev` member; the last node in the list has a NULL value stored in its `next` member. There are no bogus nodes.

Write a single function, whose argument is a pointer to a node in the list, and which deletes the node from the list and frees the memory that was used to hold that node.

Don't worry about communicating back to the caller if you happen to delete the head or the tail node.

Good code for this function appears below:

```
void delete_node(struct node *delete_me)
{
    if (delete_me->next)
        delete_me->next->prev = delete_me->prev;

    if (delete_me->prev)
        delete_me->prev->next = delete_me->next;

    free(delete_me);
}
```

If you had written this, you would have gotten a full 25 points. One student pointed out that the function is only a few lines of code, and is hardly worth 25 points, and yes, that's true—I expected it to be cheap pork. (Not quite free pork.)

In fact, only two students got the full 25 points. Most people forgot to check to see that `delete_me->next` was not NULL before they set the `prev` member of the structure it pointed to with `delete_me->next->prev = ...`. `delete_me->next` is NULL when the function is asked to delete the last node in the list; therefore most people turned in programs that failed when asked to delete the last node in a list, and in the similar case of being asked to delete the first node in a list. I wish I had stressed this more in class: Dereferencing the NULL pointer is an incredibly common mistake, which yields undefined behavior. On a good computer, a program which dereferences a NULL pointer fails immediately. On a crummy computer like an IBM PC, all that happens is that you clobber some of the operating system's data structures and cause a mysterious machine crash ten minutes down the line. Because I did not push it hard in class, people who forgot to check to see if the node was at the beginning or the end of the list lost only 5 points, although in real life they would have paid a big time penalty as they wondered why their program crashed the machine all

the time. I am sure that some people did wonder this while they worked on their calculator programs, but it is not their fault that I could not grade the assignment before the exam was due.

Some people included a handler in case the argument was `NULL`:

```
if (delete_me == NULL)
    return;
```

which is a good idea but was not required.

Some people forgot to free the memory that `delete_me` pointed to; this cost 5 points.

Some people made the function's return value an `int` instead of `void`; this would have been okay if they had actually returned a useful value to indicate success or failure,¹ but none of them did do this; this mistake cost 3 points.

¹For example, they might have passed back the return value of `free`.