2) (10 pts) ALG (Linked Lists)

Suppose we have a linked list implemented with the structure below. The function below takes in a pointer, **head**, to a linked list which is guaranteed to store data in strictly ascending order. If the list doesn't contain the value 3, the function should create a struct node storing 3 in its data component, insert the node so that the listed pointed to by head stores its data, including 3, in strictly ascending order, and returns a pointer to the front of the resulting list. If a node already exists storing 3 in the list pointed to by head, then return head and make no changes to the list.

```
typedef struct node {
    int data;
    struct node* next;
} node;

node* addValue3(node* head) {

    if ( head == NULL || head->data > 3 ) {
        node* tmp = malloc(sizeof(node));
        tmp->data = 3;
        tmp->next = head;
        return tmp;
    }

    if ( head->data == 3 )
        return head;

    node* iter = head;
    while (iter->next != NULL && iter->next->data < 3 )

        iter = iter->next;

    if ( iter->next != NULL && iter->next->data == 3 )
        return head;

    node* tmp = malloc(sizeof(node));
    tmp->data = 3;
    tmp->next = iter->next;
    iter->next = tmp ;
    return head;
}
```

Grading: 1 pt per slot, record an integer grade. If two slots are partially correct, you may just take 1 point off.