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

    if ( ____________________________ )
        return head;

    node* iter = head;

    while (iter->next != NULL && _______________________________ )

        iter = ___________________;

    if ( ____________________________ && ____________________________ )
        return head;

    node* tmp = malloc(sizeof(node));
    tmp->data = 3;

    tmp->next = ____________________________ ;

    iter->next = ____________________________ ;

    return ____________________ ;
}
```