

2) (5 pts) ALG (Linked Lists)

Consider the following function that takes in as a parameter a pointer to the front of a linked list(`list`) and the number of items in the list(`size`). *node* is defined as follows:

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

int mystery(node* list, int size) {
    node* prev = list;
    node* temp = list->next;

    while (temp != NULL) {
        if (list->data == temp->data) {
            prev->next = temp->next;
            free(temp);
            size--;
            temp = prev->next;
        }
        else {
            prev = prev->next;
            temp = temp->next;
        }
    }
    return size;
}
```

If **mystery(head, 7)**, is called, where head is shown below, what will the function return and draw a picture of the resulting list, right after the call completes?

```
+-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+
| 26 |-->| 39 |-->| 26 |-->| 20 |-->| 26 |-->| 32 |-->| 39 |-->NULL
+-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+
^ head
```

Adjusted List

```
+-----+ +-----+ +-----+ +-----+ +-----+
| 26 |-->| 39 |-->| 20 |-->| 32 |-->| 39 |-->NULL
+-----+ +-----+ +-----+ +-----+ +-----+
^ head
```

The function returns 5.

Grading: 2 pts return value (all or nothing), 3 pts list, give partial for list as you see fit.