

## 2) (5 pts) ALG (Linked Lists)

Suppose we have a linked list implemented with the structure below. We also have a function that takes in the head of the list and the current number of nodes in the list.

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

int whatDoesItDo (node * head, int size) {
    node * current = head;
    node * other;

    if (size < 2)
        return size;

    other = head->next;

    while (current != NULL) {
        current->next = other->next;
        free(other);
        current = current->next;
        size--;
        if (current != NULL && current->next != NULL) {
            current = current->next;
            other = current->next;
        }
    }

    return size;
}
```

If we call what DoesItDo(head, 8) on the following list, show the list after the function has finished and state the return value.

head -> 3 -> 8 -> 12 -> 5 -> 1 -> 7 -> 19 -> 2

Picture of List Pointed to by head After Function Call:

Head -> 3 -> 12 -> 5 -> 7 -> 19

Function Return Value: 5

**Grading: 2 pts for return value (all or nothing), 3 pts for list, give 3 pts if correct, give 2 pts if off by 1 item, 1 pt if off by 2 items, 0 otherwise**