2) (10 pts) DSN (Linked Lists)

Given a singly integer linked list, complete the following user defined function definition `moveHeadNearTail`. The user defined function moves the head node of some singly linked list that is passed to the second last position of the list (the node that comes before the tail node itself). The following figure shows a sample scenario. The function returns the head of the modified linked list. **You may assume the linked list pointed to by head has at least 3 elements in it.**

Before moveHeadNearTail

After moveHeadNearTail

```
typedef struct node_s {
    int data;
    struct node_s* next;
} node_t;

node_t * moveHeadNearTail(node_t * head) {

    node_t* tmp = head;                  // 1 pt
    while (tmp->next->next != NULL)      // 2 pts
        tmp = tmp->next;                 // 1 pt

    node_t* newfront = head->next;       // 1 pt
    head->next = tmp->next;              // 2 pts

    tmp->next = head;                    // 2 pts

    return newfront;                     // 1 pt
}
```

Grading: 4 pts for putting a temp pointer at the second to last node.
2 pts for storing and returning the new front
2 pts for linking first node next to last node
2 pts for linking second to last node to first node

Note: There were quite a few creative (correct) solutions significantly different than this one that were submitted by students during the exam. A couple of these techniques were:

(a) Storing the first value in the list in a temporary variable and copying the 2nd value into the 1st node, 3rd value into the 2nd node, etc, until getting to the second to last place and copying the temp variable into that node.

(b) Recursively moving the first node to the second slot (swapping the position of the nodes) until it's the second to last node in the list. The method has to return the new front of the list to fully work.