3) (10pts) DSN (Linked Lists)

Write a recursive function that takes the head of a linked list (possibly NULL) that contains positive integers only. The function must return -1 if the list contains any integer that is equal to the sum of all integers that come after it in the list. If not, the function can return whatever value you feel is appropriate other than -1. (Figuring out what to return is part of the fun for this problem.)

For example, the function should return -1 for the following linked list because 4 is the sum of all the nodes that follow it (1, 2, and 1):

```
20 -> 3 -> 1 -> 4 -> 1 -> 2 -> 1 -> NULL

^
head
```

The function signature and node struct are:

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

int listylist(node *head) {

   int sum;

   if (head == NULL)
      return 0;

   sum = listylist(head->next);

   if (sum == -1 || head->data == sum)
      return -1;

   return head->data + sum;
}
```

Grading:
2 pts for the base case (which should return 0 to work effectively)
2 pts for returning -1 if the recursive call itself returned -1
2 pts for returning -1 if head->data is equal to the sum from the recursive call
2 pts for returning a valid sum when not returning -1
2 pts for correct syntax and for avoiding segmentation faults

Note: There might be other solutions to this problem. Please award partial credit as necessary for alternate solutions.