

2) (10 pts) DSN (Linked Lists)

Suppose we have a stack implemented as a linked list. The stack is considered “full” if it has 20 nodes and empty if the head pointer is NULL. The nodes of the stack have the following structure:

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

Write a function to determine if the stack is full.

```
int isFull(node *stack) {  
  
    int count = 0;                //1 pt initializing a counter  
    node *helper = stack;  
  
    if (stack == NULL)            //2 pts checking if stack is null  
        return 0;  
  
    while(helper != NULL) {       //2 pts iter linked list  
        count++;                  //1 pt incrementing counter  
        helper = helper->next;    //1 pt advancing node  
    }                             // Note: can stop at 20..  
  
    if(count >= 20)               //2 pts returning true iff 20 or more  
        return 1;  
  
    return 0;                    //1 pt returning false if no  
  
    // Note: return count >= 20; takes care of both...  
}  
  
// Alternate solution.
```

```
int isFull(node* stack) {  
  
    int i;                        // 1 pt  
    for (i=0; i<20; i++) {       // 2 pts  
        if (stack == NULL) return 0; // 3 pts  
        stack = stack->next;      // 2 pts  
    }  
    return 1;                    // 2 pts  
}
```