2) (10 pts) ALG (Linked Lists)

Suppose we have a queue implemented as a doubly linked list using the structures shown below. Use head for the front of the queue and tail for the end of the queue.

```
struct node {
    int data;
    struct node* next, *prev;
}

struct queue {
     int size;
     struct node *head, *tail;
}
```

Write a dequeue function for this queue. If the queue is NULL or is already empty, return 0 and take no other action. If the queue isn't empty, dequeue the appropriate value, make the necessary adjustments, and return the dequeued value. **(Note: You must free the node that previously stored the dequeued value.)**

```
int dequeue(queue *thisQ) {

     if(thisQ == NULL)                       // Grading - 1 pt
        return 0;

     if(thisQ->size == 0)                    // Grading - 1 pt
        return 0;

     int retval = thisQ->head->data;         // Grading - 1 pt

     node *temp = thisQ->head;               // Grading - 1 pt

     thisQ->head = thisQ->head->next;        // Grading - 1 pt

     if (this->size > 1)                     // Grading - 1 pt
         thisQ->head->prev = NULL;
     else
         thisQ->tail = NULL;

     free(temp);                             // Grading - 2pts

     thisQ->size--;                          // Grading - 1 pt

     return retval;                          // Grading - 1 pt
}
```