

2) (10 pts) ALG (Linked Lists)

Suppose we have a queue implemented as a doubly linked list using the structures shown below with head pointing to node at the front of the queue and tail pointing to the node at the end of the queue.

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

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

Write an enqueue function for this queue. If the queue is already full, return 0 and take no other action. If the queue has not been created yet, return 0 and take no other action. If the queue isn't full, enqueue the integer `item` into the queue, make the necessary adjustments, and return 1. Since there is no fixed size, the queue will be considered full if a new node can't be allocated.

```
int enqueue(queue *thisQ, int item) {

    struct node *newNode = _____ ;

    if(thisQ == NULL) return 0;

    if(newNode == NULL) return 0;

    newNode->data = _____;

    newNode->next = _____;

    thisQ->size = _____;

    if(thisQ->head == NULL) {
        newNode->prev = _____;
        thisQ->head = _____;
        thisQ->tail = _____;
        return 1;
    }

    _____;

    _____;

    _____;

    return 1;
}
```