Problem 2

1) Both remove and getValue achieve O(1). For remove, only one process is being done which is removing the first element in the queue and then setting the next element as the start of the queue.

res=*start*.data;

*start*=*start*.next;

**return** res;

For getValue, it is also O(1) because all that is being done is that a loop is run once through the queue “i” times to get the correct node data.

ptr=*start*;

**for**(**int** j=0; j<i; j++)

{

ptr=ptr.next;

}

res=ptr.data;

2) add() achieves O(n) because when adding a new node, it has to be added to the end of the queue. This has to be done using a pointer, which starts at start, and then goes through the queue until all the way at the end in which the node will then be added. Therefore, the number of times the loop runs depends on n number of nodes existing in the queue to begin with. This can be seen here:

Node ptr=**new** Node();

ptr=*start*;

**while**(ptr.next!=**null**)

{

ptr=ptr.next;

}

Node d=**new** Node();

d.data=a;

ptr.next=d;