The class webpage provides code that implements several operations on reference-based lists. Extend that code to include the following additional methods:

a) Receive a reference to the head of a list h and an integer n and add n to each element in the list.
b) Receive a reference to the head of a list h and an integer n and return the nth element in the list, or -1 if the list has less than n elements.
c) Receive a reference to the head of a list h and return a reference to the node with the largest item in the list.
d) Receive a reference to the head of a list h and return a reference to the node with the smallest item in the list.
e) Receive a reference to the head of a list h and rotate the list, moving the first element to the last position, the second to the first position, and so on. So if the list contains 1 2 3 4, after rotating it, it should contain 2 3 4 1.
f) Receive a reference to the head of a list h and build a list that contains the elements in h that are in odd positions (that is, the first, third, fifth, and so on). So if h contains 3 5 6 8 9 2, your method should build and return a list that contains 3 6 9. Make sure your method does not modify the original list.
g) Repeat the previous method, but now return the elements in even positions.
h) Receive a reference to the head of the list h and a reference to the head of the list i and append i to the end of h. So if h is 1 2 3 and i is 4 5 6, after executing your method, h should be 1 2 3 4 5 6.
i) Sort the list using quicksort. Use the following pseudocode:

```java
iNode SortListB(iNode L)
if L != null
    D = L.next
    split D into two lists as follows:
    L1, containing all the elements that are smaller than L.item
    L2, containing all the elements that are greater or equal to L.item
    L1 = SortListB(L1);
    L2 = SortListB(L2);
    L.next=L2;
    L = concatenate(L1,L);
return L
```
j) Receive a reference to the head of the list h and an integer n and return the nth smallest element in the list. Use your solutions to b) and i).
k) Receive a reference to the head of the list h and an integer n and return the nth smallest element in the list, without sorting the list. Use a modification of quicksort, as described in class.

As usual, write a report describing your results, including plots showing comparative running times for various list lengths. We are particularly interested in the relative running times of methods j) and k).