In this lab you will practice with low-level reference-based lists. Your task consists of implementing three well-known sorting algorithms and applying them to randomly generated linked lists of integers.

a) Sort the list using mergesort. Use the following pseudocode:

```
iNode SortListA(iNode L)
    if L == null || L.next == null
        return L
    Split L into
        L1, containing the first, third, fifth, etc. elements of L and
        L2, containing the second, fourth, sixth, etc. elements of L.
    L1 = SortListA(L1);
    L2 = SortListA(L2);
    return merge(L1,L2); //Merge sorted lists L1 and L2 into single sorted list.
```

b) Sort the list using quicksort. Use the following pseudocode:

```
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
```

c) Sort the list using bubblesort. Use the following pseudocode:

```
iNode SortListC(iNode L)
    len = length(L)
    for int  i = 1 to len - 1
        temp=L;
        for temp=L; temp.next!=null; temp=temp.next
            if temp.item > temp.next.item
                swap temp.item and temp.next.item
    return L;
```

Use the following definition of iNode, which consists of a node containing a single item.
You must also write methods to generate a list of random integers of a user-defined length and a method to print the list. As usual, write a report describing your results, including plots showing comparative running times for the three algorithms using various list lengths.