Compare two strings represented as linked lists

Given two linked lists, represented as linked lists (every character is a node in linked list). Write a function compare() that works similar to strcmp(), i.e., it returns 0 if both strings are same, 1 if first linked list is lexicographically greater, and -1 if second string is lexicographically greater.

Examples:

```
Input: list1 = g->e->e->k->s->a
       list2 = g->e->e->k->s->b
Output: -1

Input: list1 = g->e->e->k->s->a
       list2 = g->e->e->k->s
Output: 1

Input: list1 = g->e->e->k->s
       list2 = g->e->e->k->s
Output: 0
```

C++

```
// C++ program to compare two strings represented as linked
// lists
#include<bits/stdc++.h>
using namespace std;

// Linked list Node structure
struct Node
{
    char c;
    struct Node *next;
};

// Function to create newNode in a linkedlist
Node* newNode(char c)
{
    Node *temp = new Node;
    temp->c = c;
    temp->next = NULL;
    return temp;
};

int compare(Node *list1, Node *list2)
{
    // Traverse both lists. Stop when either end of a linked
    // list is reached or current characters don't match
    while (list1 && list2 && list1->c == list2->c)
    {
        list1 = list1->next;
        list2 = list2->next;
    }

    //  If both lists are not empty, compare mismatching
    //  characters
    if (list1 && list2)
        return (list1->c > list2->c)? 1: -1;

    // If either of the two lists has reached end
    if (list1 && !list2) return 1;
    if (list2 && !list1) return -1;

    // If none of the above conditions is true, both
    // lists have reached end
    return 0;
}

// Driver program
int main()
{
    Node *list1 = newNode('g');
    list1->next = newNode('e');
    list1->next->next = newNode('e');
    list1->next->next->next = newNode('k');
    list1->next->next->next->next = newNode('s');
    list1->next->next->next->next->next = newNode('b');

    Node *list2 = newNode('g');
    list2->next = newNode('e');
    list2->next->next = newNode('e');
    list2->next->next->next = newNode('k');
    list2->next->next->next->next = newNode('s');
    list2->next->next->next->next->next = newNode('a');

    cout << compare(list1, list2);

    return 0;
}
```

Java

```
// Java program to compare two strings represented as a linked list
// Java program to compare two strings represented as a linked list

// Linked List Class
class LinkedList {

    Node head;  // head of list
    static Node a, b;

    /* Node Class */
    static class Node {

        char data;
        Node next;

        // Constructor to create a new node
        Node(char d) {
            data = d;
            next = null;
        }
    }

    int compare(Node node1, Node node2) {

        if (node1 == null && node2 == null) {
            return 1;
        }
        while (node1 != null && node2 != null && node1.data == node2.data) {
            node1 = node1.next;
            node2 = node2.next;
        }

        // if the list are diffrent in size
        if (node1 != null && node2 != null) {
            return (node1.data > node2.data ? 1 : -1);
        }

        // if either of the list has reached end
        if (node1 != null && node2 == null) {
            return 1;
        }
        if (node1 == null && node2 != null) {
            return -1;
        }
        return 0;
    }

    public static void main(String[] args) {

        LinkedList list = new LinkedList();
        Node result = null;

        list.a = new Node('g');
        list.a.next = new Node('e');
        list.a.next.next = new Node('e');
        list.a.next.next.next = new Node('k');
        list.a.next.next.next.next = new Node('s');
        list.a.next.next.next.next.next = new Node('b');

        list.b = new Node('g');
        list.b.next = new Node('e');
        list.b.next.next = new Node('e');
        list.b.next.next.next = new Node('k');
        list.b.next.next.next.next = new Node('s');
        list.b.next.next.next.next.next = new Node('a');

        int value;
        value = list.compare(a, b);
        System.out.println(value);

    }
}

// This code has been contributed by Mayank Jaiswal
```

Python

```
# Python program to compare two strings represented as
# linked lists

# A linked list node structure
class Node:

    # Constructor to create a new node
    def __init__(self, key):
        self.c = key ;
        self.next = None

def compare(list1, list2):

    # Traverse both lists. Stop when either end of linked
    # list is reached or current characters don't watch
    while(list1 and list2 and list1.c == list2.c):
        list1 = list1.next
        list2 = list2.next

    # If both lists are not empty, compare mismatching
    # characters
    if(list1 and list2):
        return 1 if list1.c > list2.c else -1

    # If either of the two lists has reached end
    if (list1 and not list2):
        return 1

    if (list2 and not list1):
        return -1
    return 0

# Driver program

list1 = Node('g')
list1.next = Node('e')
list1.next.next = Node('e')
list1.next.next.next = Node('k')
list1.next.next.next.next = Node('s')
list1.next.next.next.next.next = Node('b')

list2 = Node('g')
list2.next = Node('e')
list2.next.next = Node('e')
list2.next.next.next = Node('k')
list2.next.next.next.next = Node('s')
list2.next.next.next.next.next = Node('a')

print compare(list1, list2)

# This code is contributed by Nikhil Kumar Singh(nickzuck_007)
```

Output:

```
1
```