# 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
```

**We strongly recommend you to minimize your browser and try this yourself first.**

## 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;
}
```

Run on IDE

## Java

```
// 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
```

Run on IDE

Output:

```
1
```

Thanks to Gaurav Ahirwar for suggesting above implementation.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above



16 Comments Category: Linked Lists

## Related Posts:

- Merge two sorted linked lists such that merged list is in reverse order
- Rearrange a given linked list in-place.
- Sort a linked list that is sorted alternating ascending and descending orders?
- Select a Random Node from a Singly Linked List
- Merge Sort for Doubly Linked List
- Point to next higher value node in a linked list with an arbitrary pointer
- Swap nodes in a linked list without swapping data
- Generic Linked List in C

(Login to Rate and Mark)

1.5 Average Difficulty : **1.5/5.0**
Based on **6** vote(s)

Add to TODO List

Mark as DONE

Like Share 13 people like this.

Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.

**16 Comments** GeeksforGeeks 1 Login

Recommend Share Sort by Newest

Join the discussion…

**Devendra Patil** • 2 days ago
In JAVA:

http://code.geeksforgeeks.org/...

• Reply • Share ›

**saiteja** • 20 days ago
There is bug in this..
after scanning the common char in lists,it only compares the next char and in both the lists
example:-compare (geeksa and geeksb) it returns 'geeksb'.....
if u compare (geeksabc and geeksb) it still returns 'geeksb' as answer
thats the mistake
think about it...sorry for poor english

• Reply • Share ›

**Sai Teja** saiteja • 20 days ago
Sorry...this post is wrong...dont mind

• Reply • Share ›

**Rahul Bhambri** • a month ago
python version:
https://github.com/rbhambriiit...

• Reply • Share ›

**Pankaj Dabade** Rahul Bhambri • 7 days ago
I think if you are doing it in python then the easiest way is to convert this sequence

to tuple. Tuples are comparable.

• Reply • Share ›

**Rahul Bhambri** ➚ Pankaj Dabade • 6 days ago

Still requires you to form a tuple from the given linked list, which you shall compare.

• Reply • Share ›

**Supreeth** • 2 months ago

```
import java.util.LinkedList;

public class ComparetwoStringsLinkedList {

//Compares the firstlinkedlist with secondlinkedlist
public static boolean doesntcontain (LinkedList l1, LinkedList l2){

for ( Object c : l1){

if (l1.contains(l2)){

//If firstlist contains secondlist then return true and also print 1

System.out.println(1);

return true;

}else {

//If firstlist doesn't contain secondlist then return false and also print -1
```

see more

• Reply • Share ›

**Ishaan Arora** • 2 months ago

Java Code for the same problem

```
import java.util.*;

import java.lang.*;

public class geek

{

public static void main(String[] args)

{
```

```
//System.out.println("let us begin");

LinkedList<string> l1=new LinkedList<string>();

LinkedList<string> l2=new LinkedList<string>();

l1.add("g");
```

see more

• Reply • Share ›

**Akansh** • 2 months ago

```
public int compare(Node listA, Node listB) {
if (listA == null || listB == null) {
return 1;
}

do {
if (listA == null || listB == null) {
return 1;
}

if (listA.getValue() == listB.getValue()) {
listA = listA.getNextNode();
listB = listB.getNextNode();
} else {
return -1;
}
} while (listA != null || listB !=null);

return 0;
}
```

• Reply • Share ›

**Meenakshi** • 3 months ago

```
int compare_lists(struct Node *list_1,struct Node *list_2)

{

string s1 = "";

string s2 = "";

while(list_1&&list_2)

{

s1.push_back(list_1->c);
```

s1.push_back(list_1->c);

s2.push_back(list_2->c);

list_1=list_1->next;

list_2=list_2->next;

if((list_1!=NULL)&&(list_2!=NULL))

see more

• Reply • Share ›

**V_CODER** • 3 months ago

//recursion:

int cmp(node *head1, node *head2)

{
if(head1 == NULL && head2 !=NULL) return -1;
if(head2 == NULL && head1 !=NULL) return 1;
else if(head1->data > head2->data) return 1;
else if(head2->data > head1->data) return -1;
else if(head1 == NULL && head2== NULL) return 0;

cmp(head1->next,head2->next);
}

//pls reply if any bug in code / logic

• Reply • Share ›

**#InnerPeace** • 3 months ago

# Optimization of code

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

else (list2 && !list1) return -1;

Note :No need to check each condition as only one condition would be match at a time.

• Reply • Share ›

**Mysterious Mind** ➚ #InnerPeace • 3 months ago

return statement is there in body of if block. So later if condition won't be checked.

• Reply • Share ›

**#InnerPeace** ➚ Mysterious Mind • 3 months ago

**@Mysterious Mind** Thanks for your inputs ! Agreed upon what you said.

▲ | ▼ • Reply • Share ›

**ramswish** ➚ #InnerPeace • 3 months ago

This Doesn't work!!!
with your code it will never return 0(success case).
with working principle of if() elseif() else work

▲ | ▼ • Reply • Share ›

**Kartik** ➚ #InnerPeace • 3 months ago

This optimization doesn't seem to work. Condition (!list2 && !list1) can also be true. That is why 0 is returned at the end. Please correct me if I am wrong.

▲ | ▼ • Reply • Share ›