Download and Extract

An initial setup of files is provided to you via a shell script: <u>Download potd-q16</u>

Using a terminal, extract the initial files by running the shell script you just downloaded (you will need to navigate to the directory where you saved the file):

```
sh potd-q16.sh
```

Your files for this problem will be in the `potd-q16` directory.

The Problem

Complete the `insertSorted` function in `potd.cpp` that returns nothing and accepts a `Node **` (the head of a linked list) and a `Node *` (the node you will insert into the list) as parameters. Insert the new node into the list so the list remains in sorted order. *(If the value of the new node is already in the list you can insert it before or after the other node with the same value.)*

You should assume that the list passed into the function is sorted (but might be empty). You can assume that the `Node *` will never be `NULL` (you will always be inserting a node).

Example:

```
Current List: 1 -> 2 -> 3 -> 5
// insertSorted called with a node with value=4
Updated List: 1 -> 2 -> 3 -> 4 -> 5
```

A `main` function has been provided that tests your `insertSorted` function.

Understanding `Node**`

If you do not understand why a `Node**` is being passed in, read the following:

A `Node **` is a `Node *` passed by pointer. What would happen if we just normally passed a `Node *head` representing the head of the linked list into the function? We could still alter the node that head points to. But what if we want to change the head pointer itself? E.g., what if we wanted to insert a new node into the front of the list? If we just called the function like this: `insertSorted(head, newNode)` where head is a `Node *`, then the memory address stored in the pointer is copied into the function. The function can change the pointer, but this will only change the copy of the pointer, not the pointer that lives outside the function. To change the address stored in the pointer we need to pass by pointer. The function call would then look like: `insertNode(&head, Node)`. We also could have passed the pointer by reference, though we would have to change the function signature to be: `insertSorted(Node * & head, Node * n)`.

Upload Solution

Drop files here or click to upload.

Only the files listed below will be accepted—others will be ignored.

Files
✅ potd.cpp uploaded   Show preview ⌄

Save & Grade   Save only

Files

✅ **potd.cpp**
uploaded

Show preview ⌄

Score: 5/5 (100%)

Test Results

✓ **[1/1] Test an empty list.** ⌄

✓ **[1/1] Test insert 3.** ⌄

✓ **[1/1] Test insert 3, 9.** ⌄

✓ **[1/1] Test insert 3, 9, 1.** ⌄

✓ **[1/1] Test insert 3, 9, 1, 5.** ⌄