```cpp
// Justin Dang Student ID: 1148267
/*
singly linked list: https://en.wikipedia.org/wiki/Linked_list#Singly_linked_list and a random indian youtuber(forgot)
reference for sorting linked lists: https://www.javatpoint.com/program-to-sort-the-elements-of-the-singly-linked-list

allows user to enter "unlimited" amount of ints in a linkedlist

when the user enters -1, the loop ends and the list is sorted

the sorted list is then printed
*/
#include <iostream>
using namespace std;

class Node {
public:
    int data;                       // data in each node
    class Node* next;               // address of next node(or null/0 to define as end of Queue)
    Node(int info, Node* ptr = 0) { // Structure for each node
        data = info;
        next = ptr;
    }
};

Node* head;
class LinkedList {
public:
    void addNode(int info) {
        Node* temp = new Node(info); // new node stored in temp(note no address given to show it is last node)
        Node* x;                     // Where head will be stored to prevent altering the head node
        nodeCount++;
        if (head == 0)
            head = temp;             // if our last node is null/0 then we create a node that is the front and back
        else
        {
            x = head;
            while (x->next != 0)
                x = x->next;         // finds last node(which has 0 as its next address)
            x->next = temp;
        }
    }
    void Sort() {
        Node* node1 = head, * node2 = NULL; // the two nodes that will be compared
        int temp;                           // used to hold the data that will be
```

```
            swapped
45          if (head != NULL) {
46              while (node1 != NULL) {
47                  node2 = node1->next;                // the neighboring node set to node2
48                  while (node2 != NULL) {
49                      if (node1->data > node2->data) {// swapping if node1 is larger than node2
50                          temp = node1->data;
51                          node1->data = node2->data;
52                          node2->data = temp;
53                      }
54                      node2 = node2->next;            // allows code to loop through entire list
55                  }
56                  node1 = node1->next;                // allows code to loop through entire list
57              }
58          }
59      }
60      void print() {
61          Node* temp = head;
62          while (temp != NULL) {
63              cout << temp->data << ' ';
64              temp = temp->next;
65          }
66      }
67  private:
68      int nodeCount = 0;
69  };
70
71  int main()
72  {
73      LinkedList *list = new LinkedList();
74      int x = 0;
75      cout << "Please enter any integer: ";
76      cin >> x;
77      while (x != -1) {
78          list->addNode(x);
79          cout << "\n\nThe current list is: ";
80          list->print();
81          cout << "\n\nPlease enter any integer: ";
82          cin >> x;
83      }
84      cout << "\n\nSorted: ";
85      list->Sort();
86      list->print();
87  }
88
89  /*//--------------------------------case 1:
90  Please enter any integer: 1
91
```

```
 92
 93  The current list is: 1
 94
 95  Please enter any integer: 4
 96
 97
 98  The current list is: 1 4
 99
100  Please enter any integer: 3
101
102
103  The current list is: 1 4 3
104
105  Please enter any integer: 2
106
107
108  The current list is: 1 4 3 2
109
110  Please enter any integer: 6
111
112
113  The current list is: 1 4 3 2 6
114
115  Please enter any integer: 5
116
117
118  The current list is: 1 4 3 2 6 5
119
120  Please enter any integer: 8
121
122
123  The current list is: 1 4 3 2 6 5 8
124
125  Please enter any integer: 9
126
127
128  The current list is: 1 4 3 2 6 5 8 9
129
130  Please enter any integer: 1
131
132
133  The current list is: 1 4 3 2 6 5 8 9 1
134
135  Please enter any integer: 2
136
137
138  The current list is: 1 4 3 2 6 5 8 9 1 2
139
140  Please enter any integer: 3
141
142
143  The current list is: 1 4 3 2 6 5 8 9 1 2 3
```

```
144
145  Please enter any integer: 6
146
147
148  The current list is: 1 4 3 2 6 5 8 9 1 2 3 6
149
150  Please enter any integer: -1
151
152
153  Sorted: 1 1 2 2 3 3 4 5 6 6 8 9
154  *///--------------------------------
```