Objects and Classes

Assignment 1: List Count

Given the IntNode class, define the GetCount() function that returns the number of items in the list not including the head node.

Example 1: If the list contains: head -> 14 -> 19 -> 4

GetCount(HeadNode) returns 3

Example 2: If the list contains: head ->

GetCount(HeadNode) returns 0

main.cpp 1

main.cpp > GetCount(IntNode *)

```cpp
#include <iostream>
using namespace std;

class IntNode {
public:
   // Constructor
   IntNode(int dataInit);

   // Get node value
   int GetNodeData();

   // Get pointer to next node
   IntNode* GetNext();

   /* Insert node after this node.
      Before: this -- next
      After:  this -- node -- next
   */
   void InsertAfter(IntNode* newNode);

private:
   int dataVal;
   IntNode* nextNodePtr;
};
```

```cpp
// Constructor
IntNode::IntNode(int dataInit) {
   this->dataVal = dataInit;
   nextNodePtr = nullptr;
}

// Get node value
int IntNode::GetNodeData() {
   return this->dataVal;
}

// Get pointer to next node
IntNode* IntNode::GetNext() {
   return this->nextNodePtr;
}

/* Insert node after this node.
   Before: this -- next
   After:  this -- node -- next
*/
void IntNode::InsertAfter(IntNode* newNode) {
   IntNode* tempNext = this->nextNodePtr;
   this->nextNodePtr = newNode;
   newNode->nextNodePtr = tempNext;
}

// Return number of nodes in a list
int GetCount(IntNode* headNode) {

   /* Type your code here. */
}
```

```cpp
// Return number of nodes in a list
int GetCount(IntNode* headNode) {

   /* Type your code here. */
}

int main() {
   IntNode* headNode = new IntNode(-1);
   IntNode* currNode;
   IntNode* lastNode;

   // Initiaize head node
   lastNode = headNode;

   // Add nodes to the list
   for (int i = 0; i < 20; ++i) {
      currNode = new IntNode(i);
      lastNode->InsertAfter(currNode);
      lastNode = currNode;
   }

   cout << GetCount(headNode) << endl;

   return 0;
}
```

Assignment 1 Tests:

Apply the following 4 tests.

1. Unit test (3 points)
   Test GetCount() returns 3 for list with three items
   Show details

2. Unit test (3 points)
   Test GetCount() returns 0 for an empty list
   Show details

3. Unit test (2 points)
   Test GetCount() returns 15 for list with 15 items
   Show details

4. Unit test (2 points)
   Test GetCount() returns 1000 for list with 1000 items
   Show details

Assignment 2: Index of list item

Given the IntNode class, define the IndexOf() function to return the index of parameter target or -1 if not found.

Note: The first index after the head node is 0.

Example 1: If the list contains: head -> 14 -> 191 -> 22 -> 99

IndexOf(headNode, 22) returns 2.

Example 2: If the list contains: head ->

IndexOf(headNode, 22) returns -1.

```cpp
#include <iostream>
using namespace std;

class IntNode {
public:
   // Constructor
   IntNode(int dataInit);

   // Get node value
   int GetNodeData();

   // Get pointer to next node
   IntNode* GetNext();

   /* Insert node after this node.
      Before: this -- next
      After:  this -- node -- next
   */
   void InsertAfter(IntNode* newNode);

private:
   int dataVal;
   IntNode* nextNodePtr;
};
```

```cpp
// Constructor
IntNode::IntNode(int dataInit) {
   this->dataVal = dataInit;
   nextNodePtr = nullptr;
}

// Get node value
int IntNode::GetNodeData() {
   return this->dataVal;
}

// Get pointer to next node
IntNode* IntNode::GetNext() {
   return this->nextNodePtr;
}

/* Insert node after this node.
   Before: this -- next
   After:  this -- node -- next
*/
void IntNode::InsertAfter(IntNode* newNode) {
   IntNode* tempNext = this->nextNodePtr;
   this->nextNodePtr = newNode;
   newNode->nextNodePtr = tempNext;
}
```

```cpp
// Return index of target item
int IndexOf(IntNode* headNode, int target) {
   /* Type your code here. */

}

int main() {
   IntNode* headNode = new IntNode(-1);
   IntNode* currNode;
   IntNode* lastNode;

   // Initiaize head node
   lastNode = headNode;

   // Add nodes to the list
   for (int i = 0; i < 20; ++i) {
      currNode = new IntNode(i);
      lastNode->InsertAfter(currNode);
      lastNode = currNode;
   }

   cout << IndexOf(headNode, 15) << endl;

   return 0;
}
```

Assignment 2 Tests

Apply the following 5 tests.

1. Unit test (2 points)
   Test IndexOf() returns 2 for target in 3rd position
   Show details

2. Unit test (2 points)
   Test IndexOf() returns -1 for an empty list
   Show details

3. Unit test (2 points)
   Test IndexOf() returns 0 for target in 1st position of list
   Show details

4. Unit test (2 points)
   Test IndexOf() returns -1 if target not found
   Show details

5. Unit test (2 points)
   Test IndexOf() returns 76 for 77th time in list
   Show details

Submissions

Note: Do not forget to submit all two assignments and corresponding test outputs to receive full credit.

1 - Name your C++ files FirstName_Lastname_ListLength.cpp, FirstName_Lastname_FindIndex.cpp.

2 - Prepare your report in docx or pdf format and name it Firstname_Lastname.docx or Firstname_Lastname.pdf. Put both your assignments and corresponding tests in ONE report file.

3 - Add the screenshot of your code to the report. All tests should be performed and the result screenshot be included in the report.

Note: Make sure to have your report containing both explanatnations and screenshots.