Experiment 3(A)

Student Name: Tanmaya Kumar Pani **UID: 22BCS12986**
Branch: CSE **Section/Group: IOT-613B**
Semester: 5 **Date of Performance: 12/08/2024**
Subject Name: Advanced Programming Lab-1 **Subject Code: 22CSP-314**

1. Title: Compare two linked lists

2. Aim:

Aim: You're given the pointer to the head nodes of two linked lists. Compare the data in the nodes of the linked lists to check if they are equal. If all data attributes are equal and the lists are the same length, return .1 Otherwise, return 0

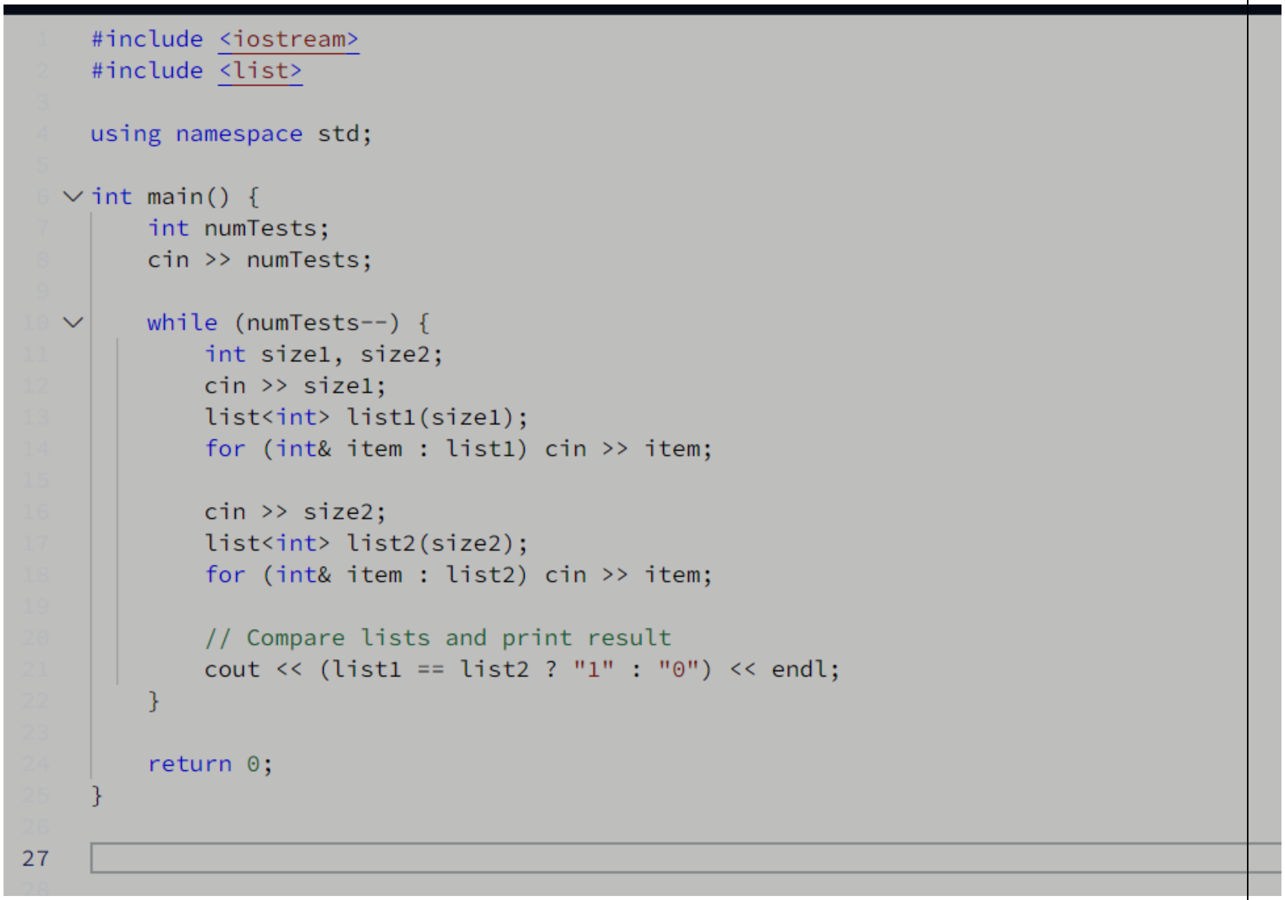
3. Objective:

compare_lists has the following parameters:
SinglyLinkedListNode llist1: a reference to the head of a list
SinglyLinkedListNode llist2: a reference to the head of a list

4. Algorithm:

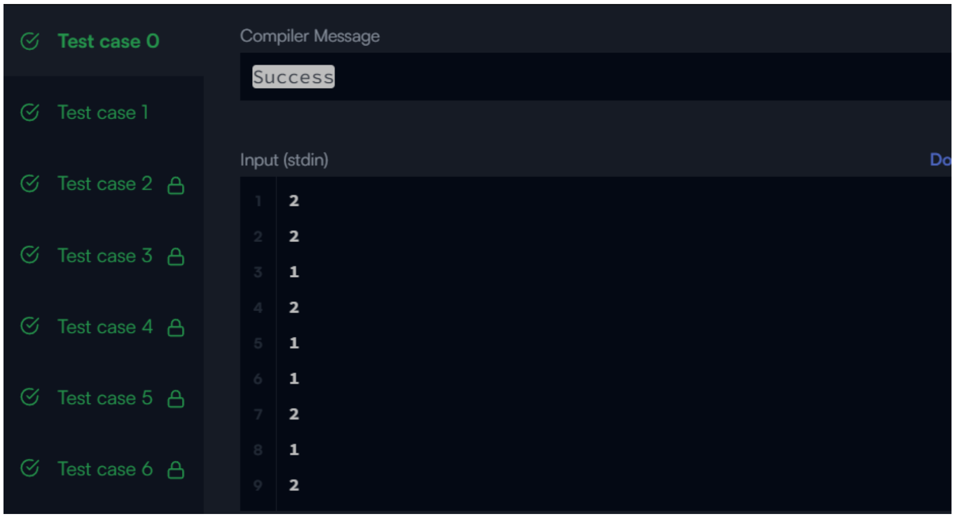
- Read number of test cases.
- Read the size of the first list and its elements.
- Read the size of the second list and its elements.
- Compare the two lists.
- Print `1` if they are identical, otherwise print `0`.
- End the program.

5. Implementation/Code



```
#include <iostream>
#include <list>

using namespace std;

int main() {
    int numTests;
    cin >> numTests;

    while (numTests--) {
        int size1, size2;
        cin >> size1;
        list<int> list1(size1);
        for (int& item : list1) cin >> item;

        cin >> size2;
        list<int> list2(size2);
        for (int& item : list2) cin >> item;

        // Compare lists and print result
        cout << (list1 == list2 ? "1" : "0") << endl;
    }

    return 0;
}
```

6. Output:



Compiler Message

```
Success
```

Input (stdin)

```
2
2
1
2
1
1
2
1
2
```

7. **Learning Outcomes:**

   - Learn to utilize std::list for managing and comparing linked lists in C++.
   - Understand how to read dynamic input sizes and data into std::list.

8. **Time Complexity: O(n1+n2)**

9. **Space Complexity: O(n1+n2)**