



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: 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.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

5. Implementation/Code

```
1  #include <iostream>
2  #include <list>
3
4  using namespace std;
5
6  int main() {
7      int numTests;
8      cin >> numTests;
9
10     while (numTests--) {
11         int size1, size2;
12         cin >> size1;
13         list<int> list1(size1);
14         for (int& item : list1) cin >> item;
15
16         cin >> size2;
17         list<int> list2(size2);
18         for (int& item : list2) cin >> item;
19
20         // Compare lists and print result
21         cout << (list1 == list2 ? "1" : "0") << endl;
22     }
23
24     return 0;
25 }
```

6. Output:

The screenshot shows a code execution environment with a sidebar on the left listing test cases from 0 to 6, each with a green checkmark. The main area is divided into two sections: 'Compiler Message' at the top, which displays 'Success' in a white box, and 'Input (stdin)' below it. The input section shows a list of 9 test cases with their respective inputs: 1: 2, 2: 2, 3: 1, 4: 2, 5: 1, 6: 1, 7: 2, 8: 1, 9: 2.

Test Case	Input (stdin)
1	2
2	2
3	1
4	2
5	1
6	1
7	2
8	1
9	2

DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

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: $O(n_1+n_2)$

9. Space Complexity: $O(n_1+n_2)$