

**UNIVERSITY INSTITUTE OF ENGINEERING**

**Department of Computer Science & Engineering**

**Subject Name:** Competitive Coding-I

**Subject Code:** 20CSP-314

|  |  |
| --- | --- |
| **Submitted By:Kanishk Soni Submitted To: Mr. Syed Abdul Basit Andrabi** | |
| **Subject Name** | CC LAB |
| **Subject Code** | 20CSP-314 |
| **Branch** | Computer Science |
| **Semester** | 5th |

**Experiment 3**

**Student Name: Kanishk Soni UID: 20BCS9398**

**Branch: CSE Section/Group:707\_WM\_B Semester: 5 Date of Performance: 9/09/2022**

**Subject Name: Competitive Coding - I Subject Code: 20CSP-314**

**Ques-1: 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 . Otherwise, return .**

**Code:**

bool compare\_lists(SinglyLinkedListNode\* head1, SinglyLinkedListNode\* head2) {

while (head1 != NULL && head2 != NULL) {

if(head1->data == head2->data){

head1 = head1->next;

head2 = head2->next;

} else {

return 0;

} }

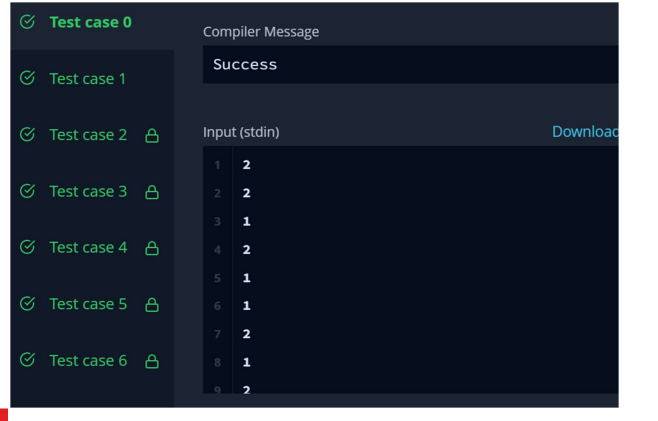
if(head1 == head2){

return true;

} else {

return false;

} }



**Ques-2: A linked list is said to contain a cycle if any node is visited more than once while traversing the list. Given a pointer to the head of a linked list, determine if it contains a cycle. If it does, return 1 Otherwise, return 0.**

Code:

bool has\_cycle(SinglyLinkedListNode\* head) {

SinglyLinkedListNode\* i = head;

SinglyLinkedListNode\* j = head;

while(j !=NULL && i !=NULL){

i = i -> next;

if(j -> next != NULL){

j = j -> next -> next;

}else{

j = NULL;

}

if(j == i && j != NULL) return true;

}

return false;

}

