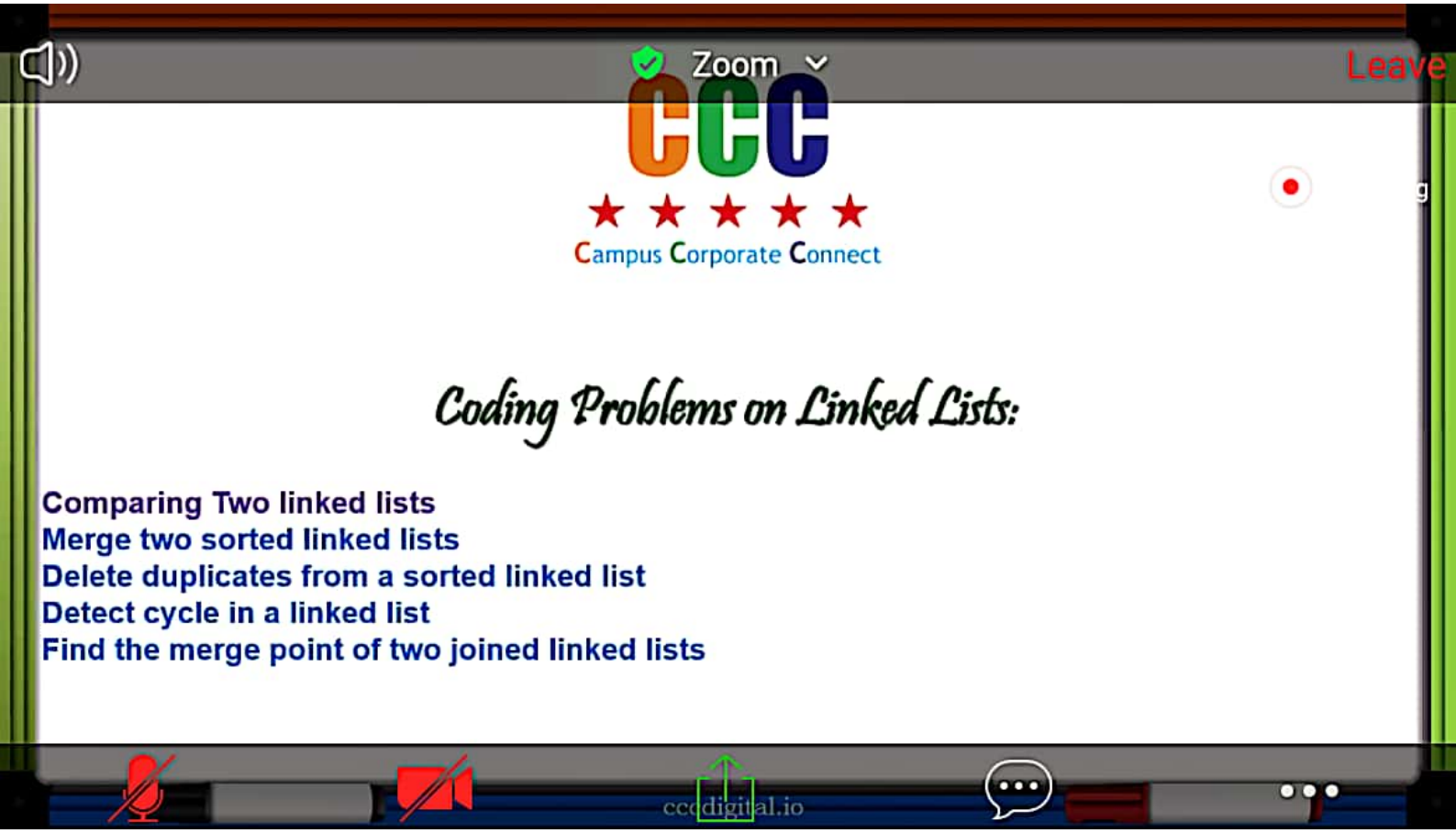
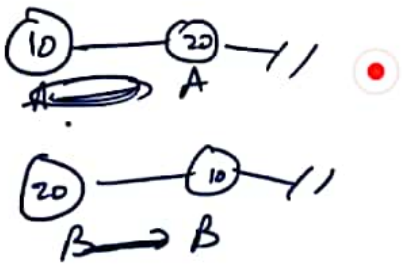

Zoom
Leave
CCC
Campus Corporate Connect
Coding Problems on Linked Lists:
Comparing Two linked lists
Merge two sorted linked lists
Delete duplicates from a sorted linked list
Detect cycle in a linked list
Find the merge point of two joined linked lists
ccdigital.io

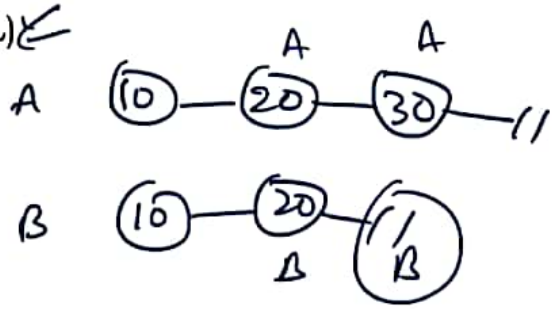
Compare two linked lists


```
int CompareLists(Node *headA, Node* headB)
{
while(headA!=NULL && headB!=NULL)
{
if(headA->data==headB->data)
{
headA=headA->next;
headB=headB->next;
}
else
return 0;
```



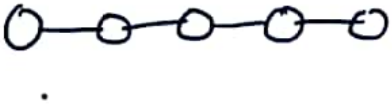
Compare two linked lists

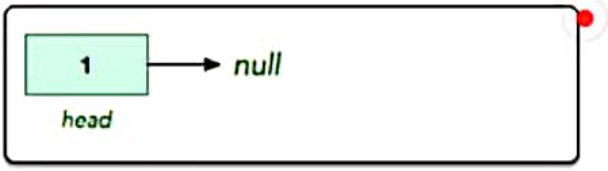

```
int CompareLists(Node *headA, Node* headB)
{
while(headA!=NULL && headB!=NULL)
{
if(headA->data==headB->data)
{
headA=headA->next;
headB=headB->next;
}
else
return 0;
}
if(headA==NULL && headB==NULL)
return 1;
else
return 0;
}
```

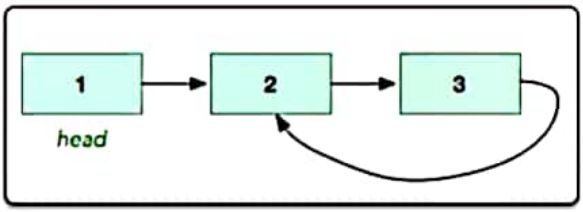


Sample Input

The following linked lists are passed as arguments to your function:







Sample Output

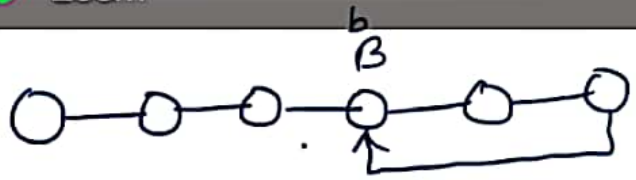
0 1

Explanation

1.The first list has no cycle, so we return *false* and the hidden code checker prints **0** to stdout.

2.The second list has a cycle, so we return *true* and the hidden code checker prints **1** to stdout.

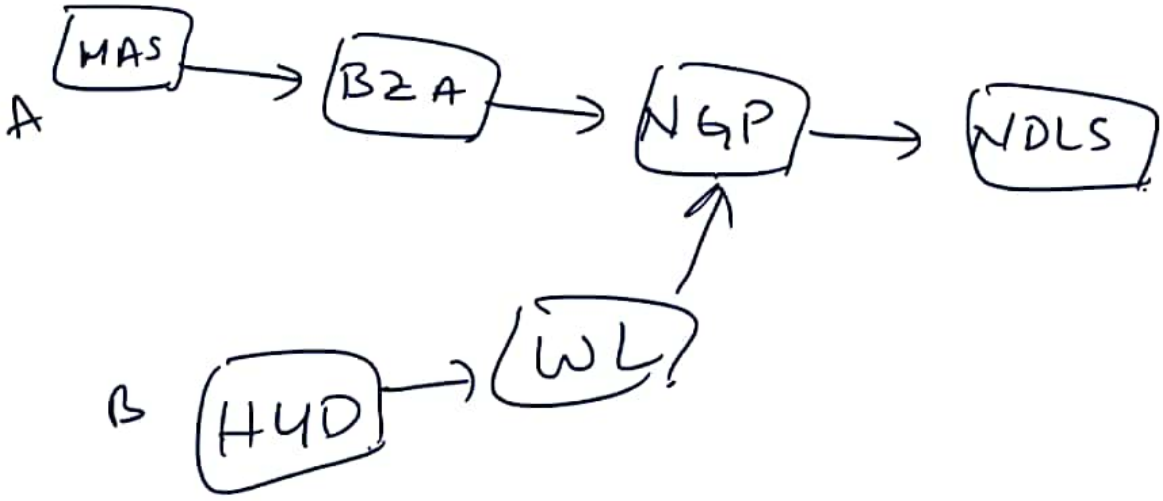
Detect cycle in linked list




```
bool has_cycle(Node* head)
{
Node* baby = head;
Node* big = head;
if(head == NULL) return 0;
do{
if(big->next == NULL || big->next->next == NULL) return 0;
big = big->next->next;
baby = baby->next;
}while(big != baby);

return 1;
```

Find Merge Point of Two Joined Linked Lists



Find Merge Point of Two Joined Linked Lists