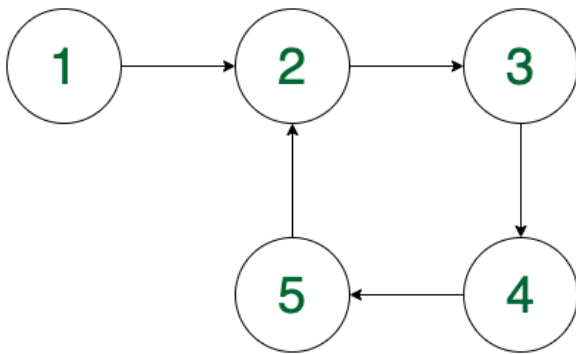


## Detect loop

Given a linked list, check if the linked list has a loop or not. The below diagram shows a linked list with a loop.



*Linked List with Loop*

### Detect loop in a linked list using Hashing:

The idea is to insert the nodes in the **hashmap** and whenever a node is encountered that is already present in the hashmap then return true.

Follow the steps below to solve the problem:

- Traverse the list individually and keep putting the node addresses in a Hash Table.
- At any point, if NULL is reached then return false
- If the next of the current nodes points to any of the previously stored nodes in Hash then return true.

Below is the implementation of the above approach:

C++

Java

Python3

C#

Javascript

```
// Java program to detect loop in a linked list
import java.util.*;
```



```
        h = h.next;
    }

    return false;
}
```

Dash



All



Articles



Videos



Problems



Quiz



Contest

```
public static void main(String[] args)
{
    LinkedList llist = new LinkedList();

    llist.push(20);
    llist.push(4);
    llist.push(15);
    llist.push(10);

    /*Create loop for testing */
    llist.head.next.next.next.next = llist.head;

    if (detectLoop(head))
        System.out.println("Loop Found");
    else
        System.out.println("No Loop");
}

// This code is contributed by Arnav Kr. Mandal.
```

## Output

Loop Found

## Detect loop in a linked list by Marking visited nodes without modifying Node structure:

### Menu

*The idea is to point the current node of the linked list to a node which is created. Whenever a node's next is pointing to that node it means loop*

Articles Read

26 of 40 Complete. (65%)

Follow the steps below to solve the problem:

- A temporary node is created.
- The next pointer of each node that is traversed is made to point to this



- Every node is checked to see if the next is pointing to a temporary node or not.
- In the case of the first node of the loop, the second time we traverse it this condition will be true, hence we find that loop exists.
- If we come across a node that points to null then the loop doesn't exist.



Below is the implementation of the above approach:

C++

Java



Quiz



Contest

```
// Java program to return first node of loop
public class GFG {

    static class Node {
        int key;
        Node next;
    };

    static Node newNode(int key)
    {
        Node temp = new Node();
        temp.key = key;
        temp.next = null;
        return temp;
    }

    // A utility function to print a linked list
    static void printList(Node head)
    {
        while (head != null) {
            System.out.print(head.key + " ");
            head = head.next;
        }
        System.out.println();
    }
}
```

Menu

Articles Read

26 of 40 Complete. (65%)

```
static boolean detectLoop(Node head)
{
    // Create a temporary node
```

 Dash  
 All

 Articles

```
// This condition is for the case
// when there is no loop
if (head.next == null) {
    return false;
}

// Check if next is already
// pointing to temp
if (head.next == temp) {
    return true;
}



// Store the pointer to the next node
// in order to get to it in the next step
Node next = head.next;

// Make next point to temp
head.next = temp;

    head = next;
}

return false;
}

// Driver code
public static void main(String args[])
{
    Node head = newNode(1);
    head.next = newNode(2);
    head.next.next = newNode(3);
    head.next.next.next = newNode(4);
    head.next.next.next.next = newNode(5);
```

 Videos  
 Problems

 Quiz  
 Contest




P



90% Money-Back!

- Courses
- Tutorials
- Jobs
- Practice
- Contests

Menu

Articles Read  
26 of 40 Complete. (65%)

```
head.next.next.next.next.next = head.next.next;

boolean found = detectLoop(head);
if (found)
```

Articles

Videos

Problems

Output



Quiz

Loop Found



**Time complexity:**  $O(N)$ . Only one traversal of the loop is needed.  
**Auxiliary Space:**  $O(1)$

Mark as Read

Report An Issue

If you are facing any issue on this page. Please let us know.

Menu