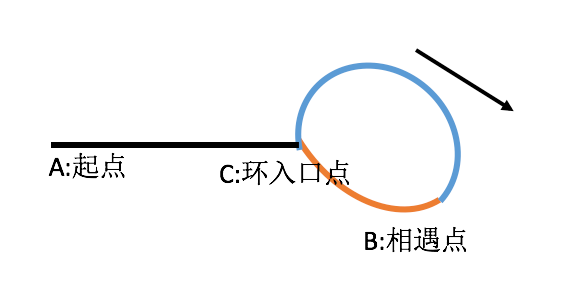
时间限制：1秒 空间限制：32768K 热度指数：71800

本题知识点： [链表](https://www.nowcoder.com/questionCenter?questionTypes=000100&mutiTagIds=580)

[算法知识视频讲解](https://www.nowcoder.com/courses/semester/algorithm-senior)

**题目描述**

一个链表中包含环，请找出该链表的环的入口结点。

思路：

快的走2步，慢的走1步，这样可以到B点相遇，而快的节点比慢的走多了一个圈。

那么快的再赋值起点的节点，然后每次走1步的话，其实由上图可以看到，蓝色和黑色的距离是一样的，那么再开始慢的和快的各走一步，就可以在C点处相遇的了。

**package** facehandjava.Linked;  
  
**public class** hasLoopDoor {  
 **public static** Node hasLoopDoor(Node node) {  
 Node node1 = node;  
 Node node2 = node;  
 **while** (node1.getNext() != **null** && node2.getNext().getNext() != **null**) {  
 node1 = node1.getNext();  
 node2 = node2.getNext().getNext();  
 **if** (node1 == node2) {  
 node2 = node;  
 **while** (node1 != node2) {  
 node1 = node1.getNext();  
 node2 = node2.getNext();  
 }  
 **return** node1;  
 }  
 }  
 **return null**;*//此为无环* }  
  
 **public static void** main(String[] args) {  
  
 Node n6 = **new** Node(11,**null**);  
 Node n5 = **new** Node(9, n6);  
 Node n4 = **new** Node(7, n5);  
 Node n3 = **new** Node(5, n4);  
 Node n2 = **new** Node(3, n3);  
 Node n1 = **new** Node(1, n2);  
*// n6.setNext(n3);* Node doorNode = *hasLoopDoor*(n1);  
 **if** (doorNode == **null**) {  
 System.***out***.println(**"链表无环"**);  
 } **else** {  
 System.***out***.println(doorNode.getVal());  
 }  
  
  
 }  
  
}