Print the Elements of a Linked List

This challenge is part of a MyCodeSchool tutorial track and is accompanied by a video lesson.

If you're new to *linked lists*, this is a great exercise for learning about them. Given a pointer to the *head* node of a linked list, print its elements in order, one element per line. If the head pointer is null (indicating the list is empty), don't print anything.

Input Format

The `void Print(Node* head)` method takes the head node of a linked list as a parameter. Each struct *Node* has a *data* field (which stores integer data) and a *next* field (which points to the next element in the list).

Note: Do not read any input from stdin/console. Each test case calls the *Print* method individually and passes it the head of a list.

Output Format

Print the integer data for each element of the linked list to stdout/console (e.g.: using *printf*, *cout*, etc.). There should be one element per line.

Sample Input

This example uses the following two linked lists:

```
NULL
1->2->3->NULL
```

$NULL$ and $Node \ 1$ are the two head nodes passed as arguments to `Print(Node* head)`.

Note: In linked list diagrams, `->` describes a pointer to the *next* node in the list.

Sample Output

```
1
2
3
```

Explanation

Test Case 0: `NULL`. An empty list is passed to the method, so nothing is printed.
Test Case 1: `1->2->3->NULL`. This is a non-empty list so we loop through each element, printing each element's data field on its own line.