Given a linked list of 0s, 1s and 2s, sort it.

Given a linked list of **N** nodes where nodes can contain values **0s**, **1s,** and **2s** only. The task is to segregate **0s**, **1s,** and **2s** linked list such that all zeros segregate to head side, 2s at the end of the linked list, and 1s in the mid of 0s and 2s.

Example 1:

```
Input:
N = 8
value[] = {1,2,2,1,2,0,2,2}
Output: 0 1 1 2 2 2 2 2
Explanation: All the 0s are segregated
to the left end of the linked list,
2s to the right end of the list, and
1s in between.
```

Example 2:

```
Input:
N = 4
value[] = {2,2,0,1}
Output: 0 1 2 2
Explanation: After arranging all the
0s,1s and 2s in the given format,
the output will be 0 1 2 2.
```

Your Task:

The task is to complete the function **segregate**() which segregates the nodes in the linked list as asked in the problem statement and returns the head of the modified linked list. The **printing** is done **automatically** by the **driver code**.

Expected Time Complexity: O(N).

Expected Auxiliary Space: O(N).

Constraints:

$1 <= N <= 10^6$