Eventual Safe States

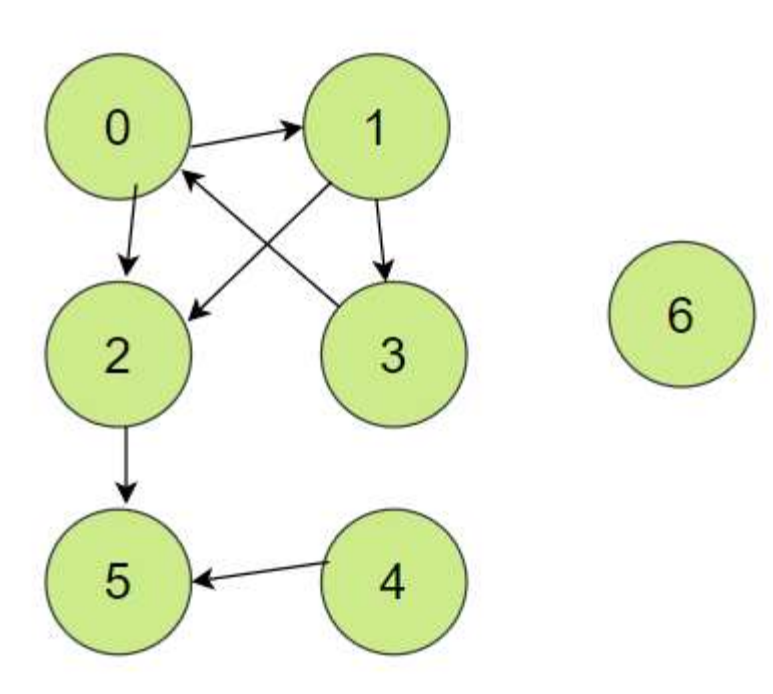
A directed graph of **V** vertices and **E** edges is given in the form of an adjacency list **adj**. Each node of the graph is labelled with a distinct integer in the range **0** to **V - 1**.

A node is a **terminal node** if there are no outgoing edges. A node is a **safe node** if every possible path starting from that node leads to a **terminal node**.

You have to return an array containing all the **safe nodes** of the graph. The answer should be sorted in **ascending** order.

Example 1:

Input:



Output:

2 4 5 6

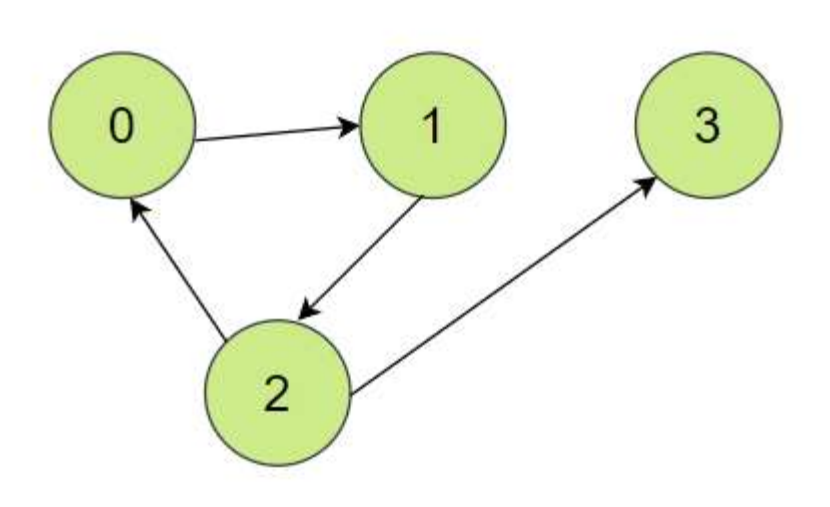
Explanation:

The given graph is shown above.

```
Nodes 5 and 6 are terminal nodes as there are no
outgoing edges from either of them.
Every path starting at nodes 2, 4, 5, and 6 all
lead to either node 5 or 6.
```

Example 2:

Input:



Output:
3
Explanation:
Only node 3 is a terminal node, and every path starting at node 3 leads to node 3.

Your Task:
You don't need to read or print anything. Your task is to complete the function **eventualSafeNodes()** which takes an integer **V** denoting no. of vertices and **adj** denoting adjacency list of the graph and returns an array of **safe nodes**.

Expected Time Complexity: O(V + E)

Expected Space Complexity: O(V)

Constraints:

- $1 <= V <= 10^4$
- $0 <= E <= 10^4$
- The graph won't contain self loops.
- Each node in the graph has a distinct value in the range 0 to V - 1.