# BFS without Queue

This program performs Breadth-First Search (BFS) traversal on a graph without using a Queue class.   
Instead, it directly uses Python lists to handle the nodes that need to be visited next.

## Explanation:

1. The graph is represented as a dictionary, where each key is a node, and its value is a list of connected nodes.  
  
2. The algorithm starts from node 'A' (the start node) and asks the user to enter a goal node.  
  
3. It uses two lists:  
 - 'a' represents the current level of nodes to explore.  
 - 'n\_l' temporarily stores nodes for the next level.  
  
4. The while loop continues until there are no more nodes left to explore.  
 - For each node in the current level ('a'), it checks if the node has already been visited.  
 - If not, the node is added to the visited list ('n\_v').  
 - If the node matches the goal, it prints the traversal path and exits.  
 - Otherwise, all child nodes are added to 'n\_l' for the next level of exploration.  
  
5. After all nodes at the current level are processed, 'a' is updated to 'n\_l', moving to the next level.  
  
6. If the goal is never found, it prints “Error.”  
  
This approach still performs BFS because it explores nodes level by level,   
but it avoids using a formal Queue class by managing lists manually.

## Example Output:

Enter your Goal : H  
Output: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']