***BFS***

* Python implementation of Depth-First Search (DFS) algorithm for graph traversal.
* Breakdown of the code
* **Initialization :**
* **graph:** This dictionary represents the graph. Keys are nodes (vertices), and their corresponding values are lists of their adjacent nodes.
* **start\_node:** This variable stores the starting node for the DFS traversal.
* **visited:** A set is used to keep track of nodes that have already been visited, preventing cycles.
* **stack:** A list (which acts as a stack using the pop() method) is used to store nodes to be explored. Initially, it contains only the start\_node.
* **DFS Traversal**
* **while stack:** The loop continues as long as there are nodes in the stack.
  + **current\_node = stack.pop():** The topmost node from the stack is popped (removed) and assigned to current\_node.
* **if current\_node not in visited:**
  + **print(current\_node, end=""):** If the current node has not been visited before, it is printed.
  + **visited.add(current\_node):** The current node is added to the visited set.
* **for neighbor in graph[current\_node]:**
  + **if neighbor not in visited:** If a neighbor of the current node has not been visited:  
    - **stack.append(neighbor):** The neighbor is pushed (added) onto the stack for exploration later.

**OUTPUT**: A D C G E F

**Made By - SWAGNIK SHIT**  
**Reg No. - 2301292134**