**Name: Muhammad Huzaifa**

**Roll No: SU92-BSAIM-S24-010**

**Section: 3A**

**Subject: AI LAB**

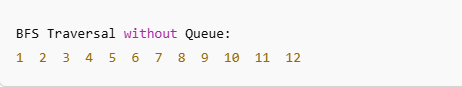
**DOCUMENTATIONS:**

**Task 1: BFS Without Queue & Without Node**

**Code Explanation:**

1. **Graph Representation:**
   * A **dictionary** is used to represent the graph.
   * Each key is a **node**, and its value is a **list of connected nodes**.
2. **Recursive BFS Implementation:**
   * **BFS\_recursive(graph, Curr\_level, visited)**
   * **Base Case:** Stops recursion when Curr\_level is empty.
   * **Processing Current Level:**
     + Iterates through all nodes at the current level.
     + If a node is **not visited**, prints it and marks it as visited.
   * **Moves to Next Level:**
     + Collects all **unvisited neighbors** into next\_level.
     + Recursively calls BFS\_recursive for the next level.
3. **Execution:**
   * Calls BFS\_recursive(graph, [1], set()) to perform BFS traversal starting from node 1.

**OUTPUT:**



**Task 2: BFS With Queue & Node**

**Code Explanation:**

1. **Node Class:**
   * Represents a **graph node** with a value and a list of neighbors (connected nodes).
2. **BFS\_use\_queue(start\_node) Function:**
   * Implements **Breadth-First Search (BFS)** using a **queue** (FIFO).
   * Initializes a queue with start\_node and an empty visited set.
   * Iterates while the queue is **not empty**:
     + **Removes the first node** from the queue.
     + **If unvisited, prints and marks it as visited.**
     + **Pushes unvisited neighbors** into the queue.
3. **Graph Construction:**
   * Creates five nodes (node\_0 to node\_4).
   * Defines their **neighbors** (connections between nodes).
4. **Execution:**
   * Calls BFS\_use\_queue(node\_0) to perform BFS traversal.

**OUTPUT:**



-------------------------------------------------------------------------------------------------------------------------------