

**Name:**  Muhammad Awais Asghar

**Roll Number:** SU92-BSAIM-F24-065

**SUBJECT:** Artifical Intelligence (LAB)

**Task No -6**

**Question 1:**

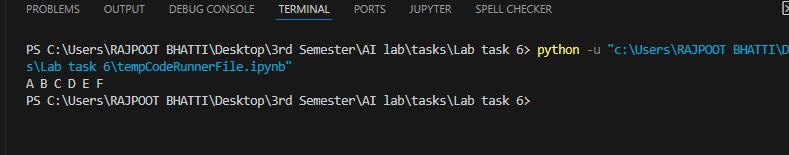
**Why this code was made:**

Without the use of a queue data structure, this code was created to carry out Breadth- First Search (BFS) on a graph.  
 Usually, a queue (FIFO) is used to implement BFS; however, in this case, a specific queue is substituted with an easy Python list (container).  
By reproducing the same behavior with list operations, it shows how BFS can still function even in the absence of the conventional queue.

**How this code works:**

**Setting up:**  
For maintaining track of which elements have already been explored, a visited set is created.The starting element is used to initialize a container list.  
  
**Looping:**Take the first element (container[0]) and remove it while the container is not empty (container = container[1:]).This simulates taking an item out of the front of a queue.  
  
**Going to:**The element is printed and added to the visited set if it hasn't been visited yet.  
  
**Including neighbors:**  
The end of the container contains all of the current element's unvisited connected elements (neighbors or "friends").This ensures that they will be investigated later, level by level, in the proper BFS order.

**Repetition:**  
The procedure is repeated until the container is empty, indicating that every element that can be reached has been investigated.



**Question 2:**

**Why this code was made:**

This code was written to use a queue to implement Breadth-First Search (BFS) on a graph.Since BFS operates in a First In, First Out (FIFO) method, the queue is the ideal data structure for one of the basic graph traversal techniques.  
This makes sure that each level of the elements is visited (all friends first, then other people, and so on).

**How this code works:**

**Setting up:**  
The elements that have been explored are recorded in a visited set.The selected starting element is where a queue list begins.  
  
**Loop Traversal:**  
Remove the first element from the queue (queue.pop(0)) while it is not empty.The earliest element added is processed first, in accordance with the FIFO principle.  
  
**Going to:**  
Print the element and mark it as visited if it hasn't been visited yet.  
  
**Queue your friend:**  
Examine every element that is directly related ("friends").Put a friend at the back of the line if they haven't been visited.This makes sure that they will receive a visit later in the proper BFS sequence.  
  
**Repetition:**  
The procedure keeps going until the queue is empty, indicating that every element that can be reached has been done so.

