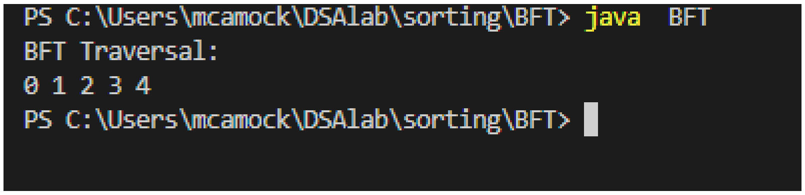
Program No:	**3**
Roll No :	**1525**
Title of Program :	
Objective :	Breadth first Traversal

SOURCE CODE:

```
/* Name: Bhairav Kedare
 * Roll: 1525
 * Program: BFT
 */

class BFT {
  private int[][] adj; // Adjacency matrix
  private boolean[] visited; // Visited nodes
  private int[] queue; // Queue for BFS
  private int front, rear; // Front and rear pointers for the queue

  // Constructor
  public BFT(int v) {
    adj = new int[v][v];
    visited = new boolean[v];
    queue = new int[v];
    front = -1;
    rear = -1;
  }

  // Add edge
  public void addEdge(int src, int dest) {
    adj[src][dest] = 1; // Set the edge from src to dest
    adj[dest][src] = 1; // Set the edge from dest to src (undirected graph)
  }

  // BFT algorithm for graph traversal
  public void performBFT(int start) {
    enqueue(start); // Enqueue starting node
    visited[start] = true; // Mark the starting node as visited

    System.out.println("BFT Traversal: ");
    while (front != -1) {
      int curr = dequeue(); // Dequeue
      System.out.print(curr + " "); // Print the current node

      // Enqueue all unvisited neighbors and set their visited flag to true
      for (int i = 0; i < adj.length; i++) {
        if (adj[curr][i] == 1 && !visited[i]) {
```

```
                    enqueue(i); // Enqueue the neighbor
                    visited[i] = true; // Mark as visited
                }
            }
        }
        System.out.println();
    }

    // Enqueue
    public void enqueue(int node) {
        if (front == -1) {
            front = 0;
        }
        rear++; // Move rear forward
        queue[rear] = node; // Add node to the queue
    }

    // Dequeue
    public int dequeue() {
        int tmp = queue[front];
        if (front == rear) {
            front = -1;
            rear = -1;
        } else {
            front++;
        }
        return tmp;
    }

    public static void main(String[] args) {
        BFT g = new BFT(6); // Create a graph with 6 vertices
        g.addEdge(0, 1);
        g.addEdge(0, 2);
        g.addEdge(0, 3);
        g.addEdge(1, 3);
        g.addEdge(2, 4);
        g.addEdge(3, 4);

        g.performBFT(0);
    }
}
```

OUTPUT:



```
PS C:\Users\mcamock\DSAlab\sorting\BFT> java  BFT
BFT Traversal:
0 1 2 3 4
PS C:\Users\mcamock\DSAlab\sorting\BFT> 
```