PROGRAM TITLE 07

BREADTH – FIRST SEARCH

AIM:

To Write the python program to implement BFS.

PROCEDURE:

1. **Initialize Data Structures:**
    - Create a queue to keep track of nodes to visit.
    - Create a set or array to keep track of visited nodes.
2. **Start at the Initial Node:**
    - Enqueue the initial node.
3. **Explore Neighbors:**
    - While the queue is not empty:
        - Dequeue a node.
        - If the node has not been visited:
            - Mark the node as visited.
            - Process the node (e.g., print its value or perform some operation).
            - Enqueue its unvisited neighbors.
4. **Termination:**
    - The algorithm terminates when the queue is empty.

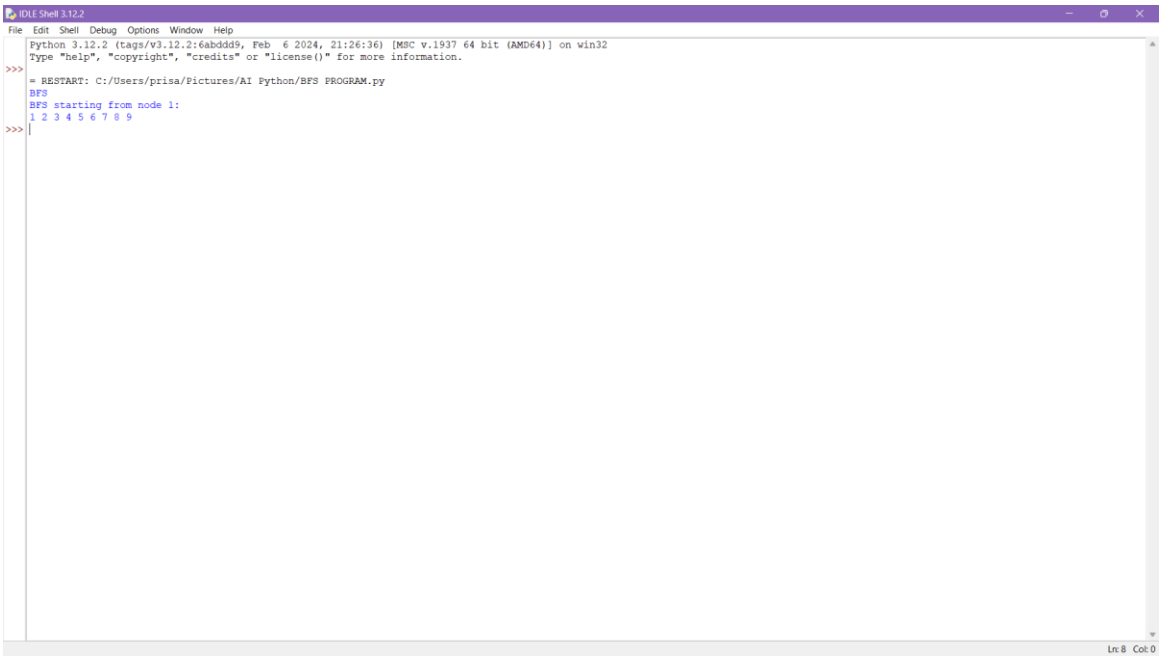
CODING:

```
from collections import deque
```

```
class Graph:     def
__init__(self):
self.graph = {}

   def add_edge(self, u, v):
if u not in self.graph:
self.graph[u] = []
self.graph[u].append(v)
def bfs(self, start):
visited = set()         queue =
deque([start])
visited.add(start)

       while queue:
           vertex = queue.popleft()
print(vertex, end=" ")

           if vertex in self.graph:                for
neighbor in self.graph[vertex]:
if neighbor not in visited:
visited.add(neighbor)
queue.append(neighbor)



if __name__ == "__main__":
   g = Graph()
   g.add_edge(0, 1)
   g.add_edge(0, 2)
```

```
g.add_edge(1, 2)
g.add_edge(2, 0)
g.add_edge(2, 3)
g.add_edge(3, 3)

print("Breadth First Traversal (starting from vertex 2):")
g.bfs(2)
```

OUTPUT:



RESULT:

Hence the program been successfully executed and verified.