Write the python program to implement BFS.

AIM

To implement **Breadth-First Search (BFS)** algorithm in Python for traversing a graph starting from a given node.

ALGORITHM

1. Represent the graph using an adjacency list.
2. Initialize a **queue** and add the start node.
3. Initialize a **visited set** to keep track of visited nodes.
4. While the queue is not empty:
    a. Dequeue a node from the front.
    b. If it has not been visited, print it and mark it as visited.
    c. Add all unvisited neighbors of the node to the queue.
5. Repeat until all reachable nodes are visited.

8 PUZZLE AI.py - C:/Users/gayathri/Downloads/8 PUZZLE AI.py (3.8.2)

File Edit Format Run Options Window Help

```
from collections import deque

def bfs(graph, start):
    visited = set()
    q = deque([start])
    while q:
        node = q.popleft()
        if node not in visited:
            print(node)
            visited.add(node)
            q.extend([n for n in graph[node] if n not in visited])

graph = {
    'A': ['B','C'],
    'B': ['D','E'],
    'C': ['F'],
    'D': [],
    'E': ['F'],
    'F': []
}

bfs(graph, 'A')
```

```
==============
A
B
C
D
E
F
>>> 
```

RESULT

The program successfully performed **Breadth-First Search (BFS)** on the given graph and printed the nodes in **level-wise order** starting from node 'A'.