Code:

```
from collections import deque

warehouse_graph = {
    'A': ['B', 'C'],
    'B': ['D', 'E'],
    'C': ['F'],
    'D': ['J'],
    'E': ['F', 'I'],
    'F': ['J']
}

def BFS(graph, start, goal):
    queue = deque([[start]])
    visited = set()
    visited.add(start)

    while queue:
        path = queue.popleft()
        current_node = path[-1]

        if current_node == goal:
            return path

        for neighbour in graph[current_node]:
            if neighbour not in visited:
                visited.add(neighbour)
                queue.append(path + [neighbour])

    return None

start_node = 'A'
goal_node = 'F'
path_found = BFS(warehouse_graph, start_node, goal_node)

if path_found:
    print(f"BFS path found from {start_node} to {goal_node}: {path_found}")
else:
    print(f"No path found from {start_node} to {goal_node}")
```

Output:

```
BFS path found from A to F: ['A', 'C', 'F']
```