Write a program to illustrate problem solving as a search.

Aim:-

To write a program to illustrate problem solving as a search

Program:-

```
def bfs_connected_component(graph, start):
    explored = []
    queue = [start]
    while queue:
        node = queue.pop(0)
        if node not in explored:
            explored.append(node)
            neighbours = graph[node]
            for neighbour in neighbours:
                queue.append(neighbour)
    return explored
def bfs_shortest_path(graph, start, goal):
    explored = []
    queue = [[start]]
    if start == goal:
        return "That was easy! start = goal"
    while queue:
        path = queue.pop(0)
        node = path[-1]
        if node not in explored:
```

```
        neighbours = graph[node]
        for neighbour in neighbours:
            new_path = list(path)
            new_path.append(neighbour)
            queue.append(new_path)
            if neighbour == goal:
                return new_path
            return "So sorry, but a connecting path doesn't exist :("
    if __name__ == '__main__':
        graph = {'A': ['B', 'C'],
                 'B': ['A', 'D'],
                 'C': ['A', 'E', 'F'],
                 'D': ['B'],
                 'E': ['C'],
                 'F': ['C'],
                }
print("\n Here's the nodes of the graph visited by", "breadth_first
      search, starting from node 'A':", bfs_connected_component(graph,'A'))
print("\n Here's the shortest path between nodes 'D' and 'F':",
      bfs_shortest_path(graph, 'D', 'F'))
```

Result:-

Thus the program to illustrate problem solving as a search has been executed successfully.