```
# Name: Aniket Sarjerao Sable
# Roll No: 54

# =============================================
# AI Practical 02-B
# Problem Statement:
# Write a program to solve the Water Jug Problem using Breadth-First Search (BFS)
# =============================================

from queue import Queue

# ---------------------------------------------
# BFS algorithm to solve the Water Jug Problem
# ---------------------------------------------
def water_jug_bfs(capacity_jug1, capacity_jug2, target):
    visited_states = set()
    q = Queue()
    q.put((0, 0))  # Starting state: both jugs are empty

    while not q.empty():
        current_state = q.get()

        if current_state in visited_states:
            continue

        visited_states.add(current_state)
        jug1, jug2 = current_state

        # Goal check: either jug has the target amount
        if jug1 == target or jug2 == target:
            return current_state

        # Possible operations:
        # 1. Fill Jug1
        q.put((capacity_jug1, jug2))
        # 2. Fill Jug2
        q.put((jug1, capacity_jug2))
        # 3. Empty Jug1
        q.put((0, jug2))
        # 4. Empty Jug2
        q.put((jug1, 0))
        # 5. Pour Jug1 → Jug2
        pour_amount = min(jug1, capacity_jug2 - jug2)
        q.put((jug1 - pour_amount, jug2 + pour_amount))
        # 6. Pour Jug2 → Jug1
        pour_amount = min(jug2, capacity_jug1 - jug1)
        q.put((jug1 + pour_amount, jug2 - pour_amount))

    return None  # No solution found

# ---------------------------------------------
# Example Usage
# ---------------------------------------------
capacity_jug1 = 5
capacity_jug2 = 3
```

```
target = 4

result = water_jug_bfs(capacity_jug1, capacity_jug2, target)
print("Water Jug Solution:", result)

# ---------------------------------------------
# Sample Output:
# Water Jug Solution: (4, 0)
# ---------------------------------------------
```