

```

from collections import deque
def water_jug_problem(capacityA, capacityB, target):
    def get_neighbors(x, y):
        return [
            ((capacityA, y), "Fill Jug1"),
            ((x, capacityB), "Fill Jug2"),
            ((0, y), "Empty Jug1"),
            ((x, 0), "Empty Jug2"),
            ((x - min(x, capacityB - y), y + min(x, capacityB - y)), "Pour Jug1 into Jug2"),
            ((x + min(y, capacityA - x), y - min(y, capacityA - x)), "Pour Jug2 into Jug1"),
        ]
    queue = deque([((0, 0), [])])
    visited = {(0, 0)}
    while queue:
        (x, y), path = queue.popleft()
        if x == target or y == target:
            return path, (x, y)
        for (next_x, next_y), action in get_neighbors(x, y):
            if (next_x, next_y) not in visited:
                visited.add((next_x, next_y))
                queue.append(((next_x, next_y), path + [(next_x, next_y, action)]))
    return None, None
def main():
    print("Water Jug Problem Solver")
    try:
        capacityA = int(input("Enter the capacity of Jug1: "))
        capacityB = int(input("Enter the capacity of Jug2: "))
        target = int(input("Enter the target amount of water: "))
        if target > max(capacityA, capacityB):
            print("Target exceeds both jugs' capacity. No solution possible.")
            return
        steps, final_state = water_jug_problem(capacityA, capacityB, target)
        if steps:
            print(f"Possible to measure {target} liters.")
            print("Steps:")
            for x, y, action in steps:
                print(f"Jug1: {x}, Jug2: {y} - {action}")
            print(f"Final state: Jug1: {final_state[0]}, Jug2: {final_state[1]}")
        else:
            print(f"Not possible to measure {target} liters with jugs of capacity {capacityA} and {capacityB}.")
    except ValueError:
        print("Please enter valid integers.")
if __name__ == "__main__":
    main()

```

Output:

Water Jug Problem Solver

Enter the capacity of Jug1: 4

Enter the capacity of Jug2: 3

Enter the target amount of water: 2

Possible to measure 2 liters.

Steps:

Jug1: 0, Jug2: 3 - Fill Jug2

Jug1: 3, Jug2: 0 - Pour Jug2 into Jug1

Jug1: 3, Jug2: 3 - Fill Jug2

Jug1: 4, Jug2: 2 - Pour Jug2 into Jug1

Final state: Jug1: 4, Jug2: 2