**Exp:3 Write the Python Program for Water Jug Problem**

**Input:**

from collections import deque

def water\_jug\_bfs(cap\_a, cap\_b, target):

    visited = set()

    queue = deque()

    queue.append((0, 0, []))

    while queue:

        a, b, path = queue.popleft()

        if (a, b) in visited:

            continue

        visited.add((a, b))

        path = path + [f"({a}, {b})"]

        if a == target or b == target:

            print("Solution found:")

            for step in path:

                print(step)

            return

        possible\_moves = [

            (cap\_a, b),      # Fill Jug A

            (a, cap\_b),      # Fill Jug B

            (0, b),          # Empty Jug A

            (a, 0),          # Empty Jug B

            (a - min(a, cap\_b - b), b + min(a, cap\_b - b)),

            (a + min(b, cap\_a - a), b - min(b, cap\_a - a))

        ]

        for move in possible\_moves:

            if move not in visited:

                queue.append((move[0], move[1], path))

    print("No solution found.")

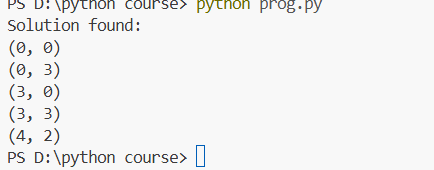
cap\_a = 4  # Jug A capacity

cap\_b = 3  # Jug B capacity

target = 2 # Target amount

water\_jug\_bfs(cap\_a, cap\_b, target)

**output:**

****