TITLE: Write the python program for Water Jug Problem

CODE:

```
from collections import deque


# Function to perform breadth-first search
def bfs(initial_state, target, jug1_capacity, jug2_capacity):
    visited = set()
    queue = deque([(0, 0, initial_state, 0)]) # Tuple format: (jug1, jug2, state, steps)

    while queue:
        jug1, jug2, state, steps = queue.popleft()
        if (jug1, jug2) == target:
            return steps
        visited.add((jug1, jug2))

        # Fill jug1
        if jug1 < jug1_capacity and (jug1_capacity, jug2) not in visited:
            queue.append((jug1_capacity, jug2, state, steps + 1))

        # Fill jug2
```

```
    if jug2 < jug2_capacity and (jug1, jug2_capacity) not in
visited:
        queue.append((jug1, jug2_capacity, state, steps + 1))

    # Empty jug1
    if jug1 > 0 and (0, jug2) not in visited:
        queue.append((0, jug2, state, steps + 1))

    # Empty jug2
    if jug2 > 0 and (jug1, 0) not in visited:
        queue.append((jug1, 0, state, steps + 1))

    # Pour from jug1 to jug2
    if jug1 > 0:
        pour_amount = min(jug1, jug2_capacity - jug2)
        if (jug1 - pour_amount, jug2 + pour_amount) not in
visited:
            queue.append((jug1 - pour_amount, jug2 +
pour_amount, state, steps + 1))

    # Pour from jug2 to jug1
    if jug2 > 0:
        pour_amount = min(jug2, jug1_capacity - jug1)
```

```
            if (jug1 + pour_amount, jug2 - pour_amount) not in
visited:
                queue.append((jug1 + pour_amount, jug2 -
pour_amount, state, steps + 1))

    return -1  # No solution found

# Main function
def main():
    jug1_capacity = int(input("Enter the capacity of jug 1: "))
    jug2_capacity = int(input("Enter the capacity of jug 2: "))
    target_amount = int(input("Enter the target amount of
water: "))

    initial_state = (0, 0)
    target_state = (0, target_amount)

    steps = bfs(initial_state, target_state, jug1_capacity,
jug2_capacity)

    if steps != -1:
        print(f"Minimum number of steps required: {steps}")
    else:
```

```
        print("Target amount cannot be achieved")


if __name__ == "__main__":
    main()
```

OUTPUT:

```
========== RESTART
Enter the capacity of jug 1: 10
Enter the capacity of jug 2: 20
Enter the target amount of water: 30
Target amount cannot be achieved
```