Exp. No: 5

Date:

Water Jug using DFS.

```
def solveWaterJugProblem (capacity-jug1, capacity-jug2, desired-quantity):
    stack = []
    stack.append ((0,0))
    while stack:
        current-state = stack.pop()
        if current-state[0] == desired-quatity or
           current-state[1] == desired-quantity:
            return current-state.
        next-states = generateNextStates (current-st
            capacity-jug1, capacity-jug2),
        stack.extend (next-states)
    return "no solution found"
def generateNextstates (state, capacity-jug1,
                        capacity-jug2):
    next-states = []
    next-states.append [(capacity-jug1, capaci
                                    jug2))
    next-states.append ((state[0], capacit
```

```
        next_states.append((0, state[1]))
        next_states.append((state[0], 0))
    pour_amount = min(state[0], capacity_jug2, state[1])
    next_state.append((state[0] - pour_amount, state[1]
    pour_amount = min(pour_amount)
                      (state[1], capacity_jug1 -
                                 state[0])
    next_states.append((state[0] + pour_amount,
                        state[1] - pour_amount))
    return next_states

solution = solve_water_jug_problem(4,3,2)
print("solution:", solution)
```

Result:-

The Program was successfully executed and the o/p is verified.