SUPERIOR UNIVERSITY

Name:

Mahak Farhan

Roll no.:

068

Class:

BSAI

Section:

4B

Subject:

Programming for Artificial Intelligence

Submitted to:

Sir Rasikh Ali

Lab 3

Water jug problem using DFS

Code:

```
def WaterJugProblem(j1_capacity, j2_capacity, target):
    visited = set()
    stack = [(0, 0, [])]
    while stack:
        j1, j2, steps = stack.pop()

        if j1 == target or j2 == target:
            print("Steps to reach the solution:")
            for step, state in steps:
                print(f"{step} --> {state}")
            return

        if (j1, j2) in visited:
            continue
        visited.add((j1, j2))

        moves = [
            (j1_capacity, j2, "Fill Jug 1"),
```

```
            (j1, j2_capacity, "Fill Jug 2"),
            (0, j2, "Empty Jug 1"),
            (j1, 0, "Empty Jug 2"),
            (max(0, j1 - (j2_capacity - j2)), min(j2 + j1, j2_capacity),
"Pour Jug 1 --> Jug 2"),
            (min(j1 + j2, j1_capacity), max(0, j2 - (j1_capacity - j1)),
"Pour Jug 2 --> Jug 1")
        ]

        for new_j1, new_j2, action in moves:
            if (new_j1, new_j2) not in visited:
                stack.append((new_j1, new_j2, steps + [(action, (new_j1,
new_j2))]))

    print("No solution found.")

j1_capacity = int(input("Enter the liters of water in Jug 1: "))
j2_capacity = int(input("Enter the liters of water in Jug 2: "))
target = int(input("Enter the target value: "))

WaterJugProblem(j1_capacity, j2_capacity, target)
```

Description:

The implemented solution uses Depth-First Search (DFS) with a stack to print all possible states to reach the target. The algorithm follows these steps:

1. Initialize a stack with the initial state (0,0) and an empty list of steps.
2. Use a loop to process each state by popping from the stack.
3. Check if the current state matches the target so, print the solution steps and exit.
4. If the state has been visited before, don't visit it again.
5. Generate all possible next states using the allowed operations.
6. Push valid, unvisited states onto the stack along with the action taken.
7. If no solution is found, print "No solution found."

Output: