

**Name :**

**Awais Ahmad**

**Roll-No:**

**SU92-BSAIM-F23-132**

**Section:**

**BSAI-4C**

**Task:**

**(3)**

---

### **Task : Water-Jug-Problem**

```
def water_jug_problem(capacityjug1, capacityjug2, goal):  
    from collections import deque  
  
    # Each state is represented as (jug1_amount, jug2_amount)  
    visited = set()  
    queue = deque()  
    parent = {} # To reconstruct the path  
  
    # Start from (0,0)  
    start_state = (0, 0)  
    queue.append(start_state)  
    visited.add(start_state)  
    parent[start_state] = None  
  
    while queue:  
        jug1, jug2 = queue.popleft()  
  
        # Check if goal is achieved  
        if jug1 == goal or jug2 == goal:  
            # Reconstruct path  
            path = []  
            current = (jug1, jug2)
```

```

        while current:
            path.append(current)
            current = parent[current]
        path.reverse()

        print("Goal achieved:", goal)
        for state in path:
            print(state)
        return True

# Generate all possible next states
possible_moves = []

# Fill jug1
possible_moves.append(((capacityjug1, jug2), "Fill jug1"))
# Fill jug2
possible_moves.append(((jug1, capacityjug2), "Fill jug2"))
# Empty jug1
possible_moves.append(((0, jug2), "Empty jug1"))
# Empty jug2
possible_moves.append(((jug1, 0), "Empty jug2"))
# Pour jug1 into jug2
transfer = min(jug1, capacityjug2 - jug2)
new_state = (jug1 - transfer, jug2 + transfer)
possible_moves.append((new_state, "Pour jug1 into jug2"))

# Pour jug2 into jug1
transfer = min(jug2, capacityjug1 - jug1)
new_state = (jug1 + transfer, jug2 - transfer)
possible_moves.append((new_state, "Pour jug2 into jug1"))

# Explore all new states
for state, action in possible_moves:
    if state not in visited:
        visited.add(state)
        parent[state] = (jug1, jug2)
        queue.append(state)

print("Solution not found")
return False

# User input
jug1Capacity = int(input("Enter the capacity of jug 1: "))
jug2Capacity = int(input("Enter the capacity of jug 2: "))
target = int(input("Enter the Goal: "))

```

```
water_jug_problem(jug1Capacity, jug2Capacity, target)
```

### Output:

```
Goal achieved: 3
```

```
(0, 0)
```

```
(3, 0)
```

```
True
```