PRINCIPLES OF ARTIFICIAL INTELLIGENCE
LABORATORY PROGRAMS

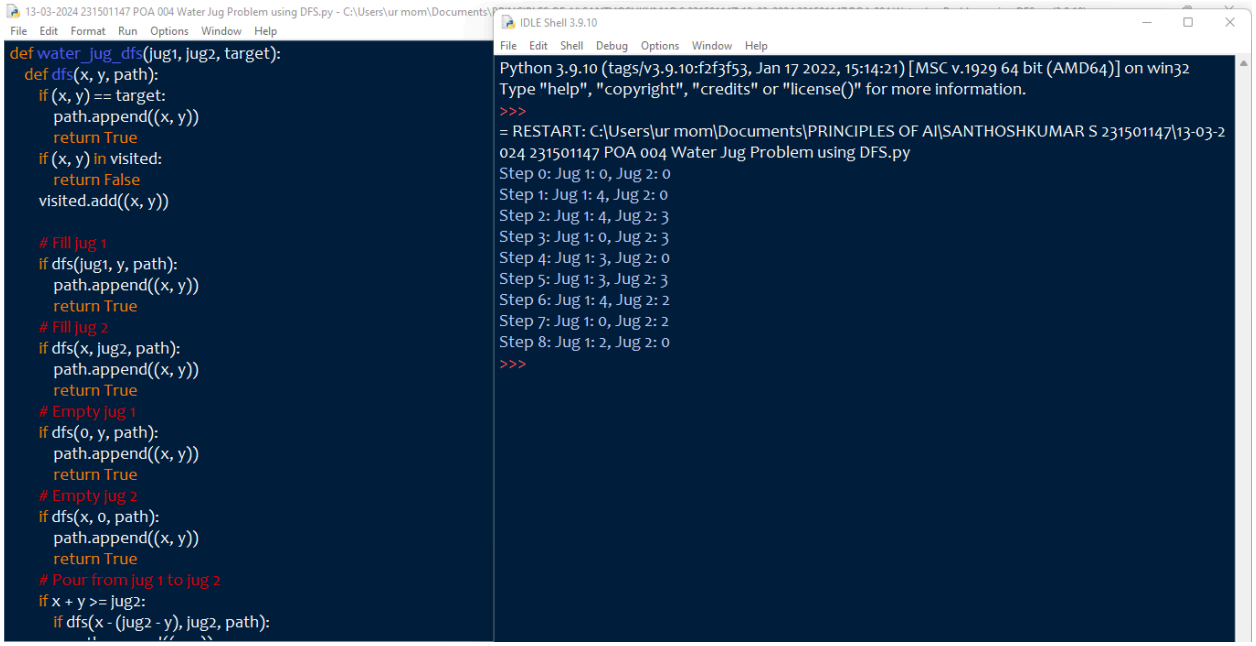
WATER JUG PROBLEM USING DEPTH FIRST SEARCH:

SOURCE CODE:

```
def water_jug_dfs(jug1, jug2, target):
    def dfs(x, y, path):
        if (x, y) == target:
            path.append((x, y))
            return True+-
        if (x, y) in visited:
            return False
        visited.add((x, y))

        # Fill jug 1
        if dfs(jug1, y, path):
            path.append((x, y))
            return True
        # Fill jug 2
        if dfs(x, jug2, path):
            path.append((x, y))
            return True
        # Empty jug 1
        if dfs(0, y, path):
            path.append((x, y))
            return True
        # Empty jug 2
        if dfs(x, 0, path):
            path.append((x, y))
            return True
        # Pour from jug 1 to jug 2
        if x + y >= jug2:
            if dfs(x - (jug2 - y), jug2, path):
                path.append((x, y))
                return True
        else:
```

```
            if dfs(0, x + y, path):
                path.append((x, y))
                return True
        # Pour from jug 2 to jug 1
        if x + y >= jug1:
            if dfs(jug1, y - (jug1 - x), path):
                path.append((x, y))
                return True
        else:
            if dfs(x + y, 0, path):
                path.append((x, y))
                return True
        return False

    visited = set()
    path = []
    if dfs(0, 0, path):
        path.reverse()
        return path
    else:
        return "No solution found."

jug1_capacity = 4
jug2_capacity = 3
target_amount = (2, 0)
solution_path = water_jug_dfs(jug1_capacity, jug2_capacity, target_amount)
if solution_path != "No solution found.":
    for step, (x, y) in enumerate(solution_path):
        print(f"Step {step}: Jug 1: {x}, Jug 2: {y}")
else:
    print("No solution found.")
```

OUTPUT:


13-03-2024 231501147 POA 004 Water Jug Problem using DFS.py - C:\Users\ur mom\Documents\
File Edit Format Run Options Window Help
def water_jug_dfs(jug1, jug2, target):
def dfs(x, y, path):
if (x, y) == target:
path.append((x, y))
return True
if (x, y) in visited:
return False
visited.add((x, y))
# Fill jug 1
if dfs(jug1, y, path):
path.append((x, y))
return True
# Fill jug 2
if dfs(x, jug2, path):
path.append((x, y))
return True
# Empty jug 1
if dfs(0, y, path):
path.append((x, y))
return True
# Empty jug 2
if dfs(x, 0, path):
path.append((x, y))
return True
# Pour from jug 1 to jug 2
if x + y >= jug2:
if dfs(x - (jug2 - y), jug2, path):
IDLE Shell 3.9.10
File Edit Shell Debug Options Window Help
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ur mom\Documents\PRINCIPLES OF AI\SANTHOSHKUMAR S 231501147\13-03-2024 231501147 POA 004 Water Jug Problem using DFS.py
Step 0: Jug 1: 0, Jug 2: 0
Step 1: Jug 1: 4, Jug 2: 0
Step 2: Jug 1: 4, Jug 2: 3
Step 3: Jug 1: 0, Jug 2: 3
Step 4: Jug 1: 3, Jug 2: 0
Step 5: Jug 1: 3, Jug 2: 3
Step 6: Jug 1: 4, Jug 2: 2
Step 7: Jug 1: 0, Jug 2: 2
Step 8: Jug 1: 2, Jug 2: 0
>>>