suma1.py - C:/Users/surya/suma1.py (3.12.2)

File Edit Format Run Options Window Help

```
def water_jug_dfs(capacity_x, capacity_y, target):
    def dfs(x, y, path):
        if x == target or y == target:
            path.append((x, y))
            return True
        if visited[x][y]:
            return False
        visited[x][y] = True
        if x < capacity_x:
            if dfs(capacity_x, y, path):
                path.append((x, y))
                return True
        if y < capacity_y:
            if dfs(x, capacity_y, path):
                path.append((x, y))
                return True
        if x > 0:
            if dfs(0, y, path):
                path.append((x, y))
                return True
        if y > 0:
            if dfs(x, 0, path):
                path.append((x, y))
                return True
        if x > 0 and y < capacity_y:
            pour = min(x, capacity_y - y)
            if dfs(x - pour, y + pour, path):
                path.append((x, y))
                return True
        if y > 0 and x < capacity_x:
            pour = min(y, capacity_x - x)
            if dfs(x + pour, y - pour, path):
                path.append((x, y))
                return True
        return False

    visited = [[False for _ in range(capacity_y + 1)] for _ in range(capacity_x + 1)]
```

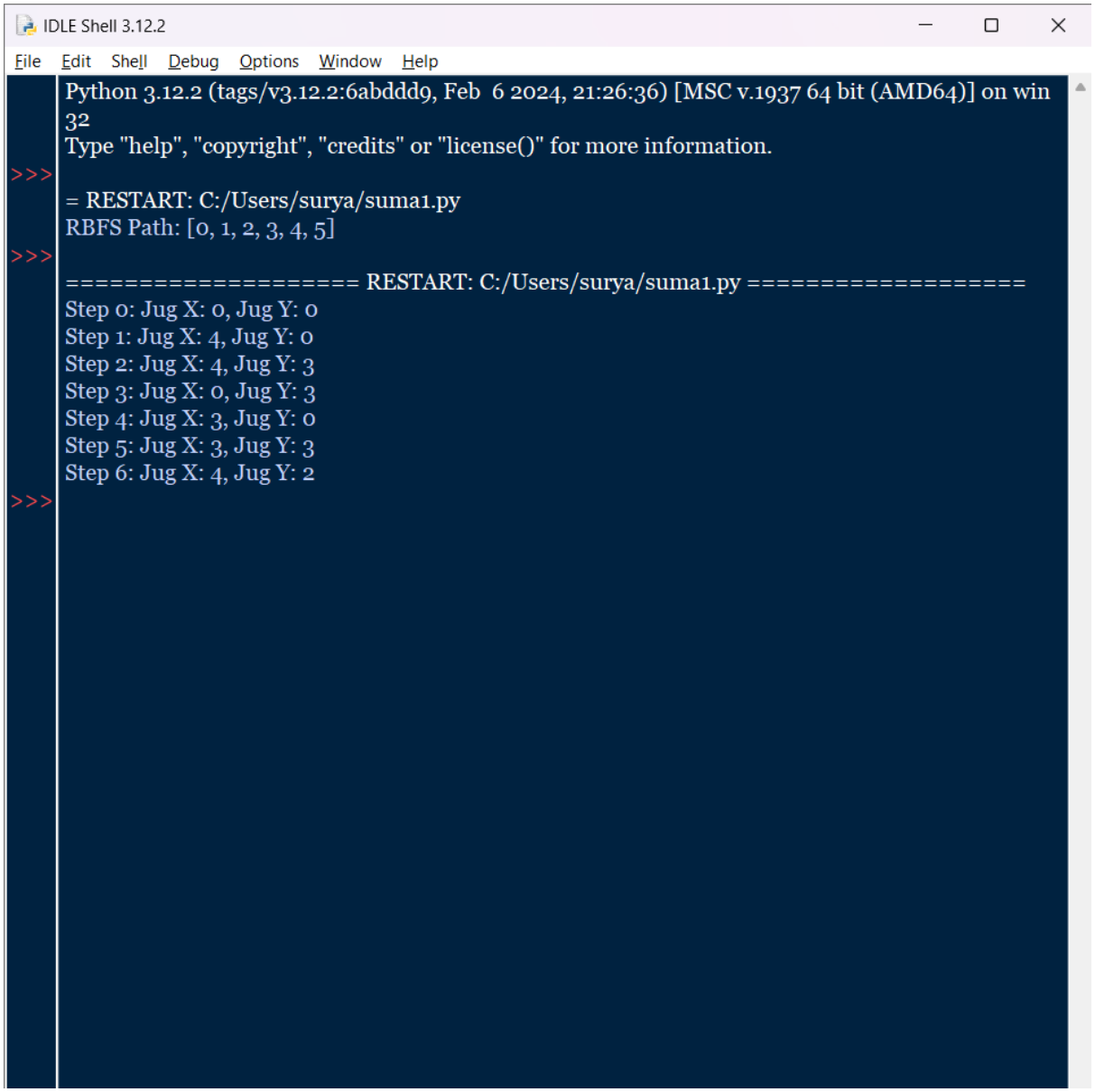
suma1.py - C:/Users/surya/suma1.py (3.12.2)

File Edit Format Run Options Window Help

```
            path.append((x, y))
            return True
        if x > 0:
            if dfs(0, y, path):
                path.append((x, y))
                return True
        if y > 0:
            if dfs(x, 0, path):
                path.append((x, y))
                return True
        if x > 0 and y < capacity_y:
            pour = min(x, capacity_y - y)
            if dfs(x - pour, y + pour, path):
                path.append((x, y))
                return True
        if y > 0 and x < capacity_x:
            pour = min(y, capacity_x - x)
            if dfs(x + pour, y - pour, path):
                path.append((x, y))
                return True
        return False

    visited = [[False for _ in range(capacity_y + 1)] for _ in range(capacity_x + 1)]
    path = []
    if dfs(0, 0, path):
        path.reverse()
        return path
    else:
        return "No solution found."

capacity_x = 4
capacity_y = 3
target = 2
solution_path = water_jug_dfs(capacity_x, capacity_y, target)
if solution_path != "No solution found.":
    for step, (x, y) in enumerate(solution_path):
        print(f"Step {step}: Jug X: {x}, Jug Y: {y}")
else:
```


IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win 32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/surya/suma1.py
RBFS Path: [0, 1, 2, 3, 4, 5]
>>>
========== RESTART: C:/Users/surya/suma1.py ==========
Step 0: Jug X: 0, Jug Y: 0
Step 1: Jug X: 4, Jug Y: 0
Step 2: Jug X: 4, Jug Y: 3
Step 3: Jug X: 0, Jug Y: 3
Step 4: Jug X: 3, Jug Y: 0
Step 5: Jug X: 3, Jug Y: 3
Step 6: Jug X: 4, Jug Y: 2
>>>