THREE WATER JUGS PROBLEM :

```
#3 water jug problem
#capacity = (8, 5, 3)
x = int(input("Enter the maximum capacity of jug A: "))
y = int(input("Enter the maximum capacity of jug B: "))
z = int(input("Enter the maximum capacity of jug C: "))
goalX = int(input("Enter the goal capacity in Jug A: "))
goalY = int(input("Enter the goal capacity in Jug B: "))
visited = {}
ans = []

def get_all_states(state):
    a = state[0]
    b = state[1]
    c = state[2]
    if a == goalX and b == goalY:
        ans.append(state)
        return True
    if (a, b, c) in visited:
        return False
    visited[(a, b, c)] = 1

    if a > 0:
        if (a + b) <= y:
            if get_all_states((0, a + b, c)):
                ans.append(state)
                return True
        else:
            if get_all_states((a - (y - b), y, c)):
                ans.append(state)
                return True
        if (a + c) <= z:
            if get_all_states((0, b, a + c)):
                ans.append(state)
                return True
        else:
            if get_all_states((a - (z - c), b, z)):
                ans.append(state)
                return True
    if b > 0:
        if (a + b) <= x:
            if get_all_states((a + b, 0, c)):
                ans.append(state)
                return True
        else:
            if get_all_states((x, b - (x - a), c)):
                ans.append(state)
                return True
        if (b + c) <= z:
            if get_all_states((a, 0, b + c)):
                ans.append(state)
                return True
        else:
            if get_all_states((a, b - (z - c), z)):
```

```
                ans.append(state)
                return True
    if c > 0:
        if (a + c) <= x:
            if get_all_states((a + c, b, 0)):
                ans.append(state)
                return True
        else:
            if get_all_states((x, b, c - (x - a))):
                ans.append(state)
                return True
        if (b + c) <= y:
            if get_all_states((a, b + c, 0)):
                ans.append(state)
                return True
        else:
            if get_all_states((a, y, c - (y - b))):
                ans.append(state)
                return True
    return False

initial_state = (8, 0, 0)
print("Solution : \n")
get_all_states(initial_state)
ans.reverse()
for i in ans:
    print(i)
```

OUTPUT :

```
Enter the maximum capacity of jug A: 8
Enter the maximum capacity of jug B: 5
Enter the maximum capacity of jug C: 3
Enter the goal capacity in Jug A: 4
Enter the goal capacity in Jug B: 4
Solution :

(8, 0, 0)
(3, 5, 0)
(0, 5, 3)
(5, 0, 3)
(5, 3, 0)
(2, 3, 3)
(2, 5, 1)
(7, 0, 1)
(7, 1, 0)
(4, 1, 3)
(4, 4, 0)
```