class State:

def \_\_init\_\_(self, jug1, jug2):

self.jug1 = jug1

self.jug2 = jug2

def \_\_eq\_\_(self, other):

return self.jug1 == other.jug1 and self.jug2 == other.jug2

def \_\_hash\_\_(self):

return hash((self.jug1, self.jug2))

def \_\_str\_\_(self):

return f"({self.jug1}, {self.jug2})"

def water\_jug\_dfs(capacity\_jug1, capacity\_jug2, target):

stack = [(State(0, 0), [])] # Initial state with empty jugs and no actions

visited = set()

while stack:

current\_state, actions = stack.pop()

# Check if we've reached the target state

if current\_state.jug1 == target or current\_state.jug2 == target:

return actions

if current\_state in visited:

continue

visited.add(current\_state)

# Fill jug 1

stack.append((State(capacity\_jug1, current\_state.jug2), actions + ["Fill jug 1"]))

# Fill jug 2

stack.append((State(current\_state.jug1, capacity\_jug2), actions + ["Fill jug 2"]))

# Empty jug 1

stack.append((State(0, current\_state.jug2), actions + ["Empty jug 1"]))

# Empty jug 2

stack.append((State(current\_state.jug1, 0), actions + ["Empty jug 2"]))

# Pour jug 1 to jug 2

pour\_to\_jug2 = min(current\_state.jug1, capacity\_jug2 - current\_state.jug2)

stack.append((State(current\_state.jug1 - pour\_to\_jug2, current\_state.jug2 + pour\_to\_jug2),

actions + [f"Pour jug 1 to jug 2 ({pour\_to\_jug2})"]))

# Pour jug 2 to jug 1

pour\_to\_jug1 = min(current\_state.jug2, capacity\_jug1 - current\_state.jug1)

stack.append((State(current\_state.jug1 + pour\_to\_jug1, current\_state.jug2 - pour\_to\_jug1),

actions + [f"Pour jug 2 to jug 1 ({pour\_to\_jug1})"]))

return "No Solution Found"

# Input capacities and target amount

jug1\_capacity = int(input("Enter Jug 1 capacity: "))

jug2\_capacity = int(input("Enter Jug 2 capacity: "))

target\_amount = int(input("Enter target amount: "))

# Perform DFS to find solution

result = water\_jug\_dfs(jug1\_capacity, jug2\_capacity, target\_amount)

# Print the result

if result:

print("Solution found:")

for action in result:

print(action)

else:

print("No solution found.")