EXP:NO:03 NAME:MOHAMMED FAISAL S.M

REG:NO:231801105

DEPT:AIDS-B

WATER JUG PROGRAM USING DFS

Program:

from collections import deque

def DFS(a, b, target):

m = {}

isSolvable = False

path = []

# Using a list as a stack for DFS

stack = []

stack.append((0, 0))

while len(stack) > 0:

u = stack.pop() # Use pop to get the last element (depth-first)

if (u[0], u[1]) in m:

continue

if u[0] > a or u[1] > b or u[0] < 0 or u[1] < 0:

continue

path.append([u[0], u[1]])

m[(u[0], u[1])] = 1

if u[0] == target or u[1] == target:

isSolvable = True

if u[0] == target and u[1] != 0:

path.append([u[0], 0])

elif u[1] == target and u[0] != 0:

path.append([0, u[1]])

sz = len(path)

for i in range(sz):

print("(", path[i][0], ",", path[i][1], ")")

return # Exiting the function after finding the solution

# Fill Jug1

stack.append((a, u[1]))

# Fill Jug2

stack.append((u[0], b))

# Empty Jug1

stack.append((0, u[1]))

# Empty Jug2

stack.append((u[0], 0))

# Pour water from Jug1 to Jug2 until Jug2 is full or Jug1 is empty

pour = min(u[0], b - u[1])

stack.append((u[0] - pour, u[1] + pour))

# Pour water from Jug2 to Jug1 until Jug1 is full or Jug2 is empty

pour = min(u[1], a - u[0])

stack.append((u[0] + pour, u[1] - pour))

if not isSolvable:

print("No solution")

if \_\_name\_\_ == '\_\_main\_\_':

Jug1, Jug2, target = 4, 3, 2

print("Path from initial state to solution state:")

DFS(Jug1, Jug2, target)

Output:

