Dater Jug to water Ity DFS (jug) - capractel. Jug) - capracty, tright current state . Nondo orsited = To implement DFS algorithm to water jug merion state is Mone covered state = (0,0) Algorithm 1. Start 2. Initiale starting starte with 3. Mark current state as visited 4. If amount of water is equal in both jugs print the solution and terminate 5. Generate all possible next states 6. Recursively apply DFS to curvisited state 7. Stop Just Just - capacity Just Hand Long - long) ware 2 bolyun (Sont - harvey - 2 hal - 1 bol (0 2) xou.) (25 Br P. 1 Br. P for restate in possible moves if water Ing DFS (jugl - capacity, jug2-capacity larget, new neartable, verbed):

surtron Tome

```
def water Tug DFS (jug1 - capacity, jug2 - capacity, target
                        convert-state = None, visited = None).
    pui, réfor visited is None: 270 landque
               visited = sel()
             current_state is None:
                curvent-state = (0,0)
                                                 Maritim
             current_state in visited:
                                      1. Stant
                 return False;
          visiled. add (current-state)
          jug 1 jug 2 = avvient-state
          point (f" Tugl: fjug 13, Tug2: fjug 23)
           if jugl == Earget on jug2 == Larget
  pount ("Solution found!
Tour Tours I bretween True
    estada test voltaron la staranon. à
            (jug)_capacity, jug2)
(jug)_jug2_capacity)
             (0, jug2),
                                          7 Stop
            (jug1,0)
              (min (jugl_capacity, jugl + jug2), max (0, jug2 - jugl_capocity
             (maxly 0, jug1 - (jug2-capacity - jug2), min(jug 2-capacity,
         for next state in possible_moves:
              if water Tug D=S (jug1 - capacity, jug2-capacity,
                        target, men next-state, visited):
                              return True
```

jug2_capacity = 1,

jug2_capacity = 3

target = 2

point (" DFS Toraversal : ")

if not water TugDFS (jug1_capacity, jug2_capacity,

point (" No solution found")

Output:

DFS Traversal

Jugl: 0 Jug2: 01 ali bro

Jug1: 1

Leval M. Jug2: 3 and E

Jug1: 1

Jug1: 0

Jug2: 3

Result:

The program has been executed successfully and the output has been verified

T. Stop.