**Brief Intro:** In the **water jug problem in Artificial Intelligence**, we are provided with two jugs: one having the capacity to hold 3 gallons of water and the other has the capacity to hold 4 gallons of water.

There is no other measuring equipment available and the jugs also do not have any kind of marking on them.

So, the agent’s task here is to fill the 4-gallon jug with 2 gallons of water by using only these two jugs and no other material. Initially, both our jugs are empty.

**Implementation of program to count minimum number of steps required to measure ‘d’ litre water using jugs of ‘m’ liters and ‘n’ liters capacity.**

def gcd(a, b):

if b==0:

return a

return gcd(b, a%b)

def Pour(toJugCap, fromJugCap, d):

# Initialize current amount of water

# in source and destination jugs

fromJug = fromJugCap

toJug = 0

# Initialize steps required

step = 1

while ((fromJug is not d) and (toJug is not d)):

# Find the maximum amount that can be

# poured

temp = min(fromJug, toJugCap-toJug)

# Pour 'temp' liter from 'fromJug' to 'toJug'

toJug = toJug + temp

fromJug = fromJug - temp

step = step + 1

if ((fromJug == d) or (toJug == d)):

break

# If first jug becomes empty, fill it

if fromJug == 0:

fromJug = fromJugCap

step = step + 1

# If second jug becomes full, empty it

if toJug == toJugCap:

toJug = 0

step = step + 1

return step

# Returns count of minimum steps needed to

# measure d liter

def minSteps(n, m, d):

if m> n:

temp = m

m = n

n = temp

if (d%(gcd(n,m)) is not 0):

return -1

# Return minimum two cases:

# a) Water of n liter jug is poured into

# m liter jug

return(min(Pour(n,m,d), Pour(m,n,d)))

# Driver code

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

n = 3

m = 5

d = 4

print('Minimum number of steps required is', minSteps(n, m, d))

**OUTPUT:**

Minimum number of steps required is 6

**Apparent Solution of water jug problem according to the production rules:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **4 gallon jug contents** | **3 gallon jug contents** | **Rule followed** |
| 1. | 0 gallon | 0 gallon | Initial state |
| 2. | 0 gallon | 3 gallons | Rule no.2 |
| 3. | 3 gallons | 0 gallon | Rule no. 9 |
| 4. | 3 gallons | 3 gallons | Rule no. 2 |
| 5. | 4 gallons | 2 gallons | Rule no. 7 |
| 6. | 0 gallon | 2 gallons | Rule no. 5 |
| 7. | 2 gallons | 0 gallon | Rule no. 9 |

On reaching the 7th attempt, we reach a state which is our goal state. Therefore, at this state, our problem is solved.