Write the python program for Water Jug Problem

AIM

To implement a Python program to solve the **Water Jug Problem** using the **Breadth-First Search (BFS)** technique.

ALGORITHM

1. Start with two jugs of given capacities x and y, and a target volume.
2. Represent the state as (a, b) where a = water in Jug1 and b = water in Jug2.
3. Initialize the state as (0, 0) (both empty).
4. Use a queue to explore states in BFS order.
5. For each state (a, b), generate possible next states by performing the following operations:
    a. Fill Jug1 fully.
    b. Fill Jug2 fully.
    c. Empty Jug1.
    d. Empty Jug2.
    e. Pour water from Jug1 → Jug2.
    f. Pour water from Jug2 → Jug1.
6. Mark visited states to avoid repetition.
7. Stop when either jug contains the target amount.

8 PUZZLE AI.py - C:/Users/gayathri/Downloads/8 PUZZLE AI.py (3.8.2)

File Edit Format Run Options Window Help

```
from collections import deque

def water_jug(x, y, target):
    visited = set()
    q = deque([(0, 0)])
    while q:
        a, b = q.popleft()
        if (a, b) in visited:
            continue
        visited.add((a, b))
        print(a, b)
        if a == target or b == target:
            print("Reached target!")
            return
        q.extend([
            (x, b),
            (a, y),
            (0, b),
            (a, 0),
            (a - min(a, y-b), b + min(a, y-b)),
            (a + min(b, x-a), b - min(b, x-a))
        ])
water_jug(4, 3, 2)
```

```
==========
0 0
4 0
0 3
4 3
1 3
3 0
1 0
3 3
0 1
4 2
Reached target!
>>>
```

RESULT

The program successfully solved the **Water Jug Problem** using BFS.
For jugs of capacity **4L and 3L** with target **2L**, the solution was reached at state (4, 2).