ASSIGNMENT 2

Name:

Roll No:

1)Water Jug Problem in AI:

The water Jug Problem, as the name suggests, is a problem where two jugs of water are given, say one is a 4-litre one, and the other one is a 3-litre one, but none of the measuring markers is mentioned on any of it. There is a pump available to fill the jugs with water. How can you exactly pour 2 litres of water into a 4-litre jug? Assuming that both the jugs are empty, the task is to find a solution to pour 2-litre water into a 4-litre jug.

Production Rules for Water Jug Problem:

For this problem, the state is represented as a pair (jug1, jug2), where jug1 is the amount of water in the first jug and jug2 is the amount in the second jug.

Using these production rules, we can construct a solution path to move from the initial state to the goal state.

Algorithm to Solve Water Jug Problem

n production rules for the water jug problem, let x denote a 4-litre jug, and y denote a 3-litre jug, i.e. x=0,1,2,3,4 or y=0,1,2,3

Start state (0,0)

Goal state (2,n) from any n

Start from the start state and end up at the goal state. Production rules for the water jug problem in AI are as follows:

1.	(x,y) is X<4 -> (4, Y)	Fill the 4-litre jug
2.	(x, y) if Y<3 -> (x, 3)	Fill the 3-litre jug

3.	(x, y) if x>0 -> (x-d, d)	Pour some water from a 4-litre jug
4.	(x, y) if Y>0 -> (d, y-d)	Pour some water from a 3-litre jug
5.	(x, y) if x>0 -> (0, y)	Empty 4-litre jug on the ground
6.	(x, y) if y>0 -> (x,0)	Empty 3-litre jug on the ground
7.	(x, y) if X+Y >= 4 and y>0 -> (4, y-(4-x	Pour water from a 3-litre jug into a 4-litre jug until it is full
8.	(x, y) if X+Y>=3 and x>0 -> (x-(3-y), 3))	Pour water from a 3-litre jug into a 4-litre jug until it is full
9.	(x, y) if X+Y <=4 and y>0 -> (x+y, 0)	Pour all the water from a 3-litre jug into a 4-litre jug
10.	(x, y) if X+Y<=3 and x>0 -> (0, x+	Pour all the water from a 4-litre jug into a 3-litre jug
11.	(0, 2) -> (2, 0)	Pour 2-litre water from 3-litre jug into 4-litre jug
12.	(2, Y) -> (0, y)	Empty 2-litre in the 4-litre jug on the ground.

2)Missionaries and Cannibals

Problem:

Three missionaries and three cannibals, along with one boat that fits at most two people (and requires at least one for operation), are on the left bank of a river. The most salient thing about missionaries and cannibals in "cohabitation" is that if ever the cannibals in any one spot (left bank, right bank, on the boat outnumber the missionaries, the outnumbered missionaries will be consumed – eaten! The goal of this problem is to get all six individuals safely across the river from the left bank to the right bank.

3) N Queen problems:

The core challenge of the n-queens problem is to place n queens on an n x n chessboard such that no two queens can attack each other. This means no two queens can be in the same row, column, or diagonal.

This problem is significant in the field of combinatorial optimization and has applications in various fields such as algorithm design, artificial intelligence, and operations research.

- Start in the leftmost column
- If all queens are placed return true
- Try all rows in the current column. Do the following for every row.
    - If the queen can be placed safely in this row
        - Then mark this **[row, column]** as part of the solution and recursively check if placing queen here leads to a solution.
        - If placing the queen in **[row, column]** leads to a solution then return **true**.
        - If placing queen doesn't lead to a solution then unmark this **[row, column]** then backtrack and try other rows.

- If all rows have been tried and valid solution is not found return **false** to trigger backtracking.

SET A:

1. Write a program to demonstrate Water jug problem

SET B:

2. Write a program to demonstrate Missionaries and Cannibal Problem

SET C:

3. Demonstrate N-Queen problem