Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

SUB: AIML
Branch: COMPS B

Name :	**Chetan Deepak Patil**
UID :	**2023301012**
Exp no :	**01**
Aim :	**To implement an Intelligent agent (problem formulation and implementation).**
Thesis :	

PEAS for the Missionaries and Cannibals Problem

Performance Measure

- **Objective**: Successfully transport all three missionaries and three cannibals from the left bank of the river to the right bank without any missionaries being outnumbered and eaten by cannibals on either side of the river.
- **Criteria for Success**:
  - All six individuals (three missionaries and three cannibals) must

end up on the right bank.
    - At no point should missionaries be outnumbered by cannibals on either bank of the river.
    - The number of steps (or moves) taken to achieve the goal might also be a measure of efficiency, where fewer steps are preferred.

Environment

- **State**: The environment consists of two riverbanks (left and right) and a boat. Each bank can have a different number of missionaries and cannibals.
    - **State Variables**:
        - The number of missionaries and cannibals on the left bank.
        - The number of missionaries and cannibals on the right bank.
        - The position of the boat (on the left or right bank).
    - **Constraints**:
        - On each bank, if there are missionaries present, they must not be outnumbered by cannibals.
        - The boat can carry a maximum of two people at a time.
        - The boat must always have at least one person or be empty when crossing.

Actuators

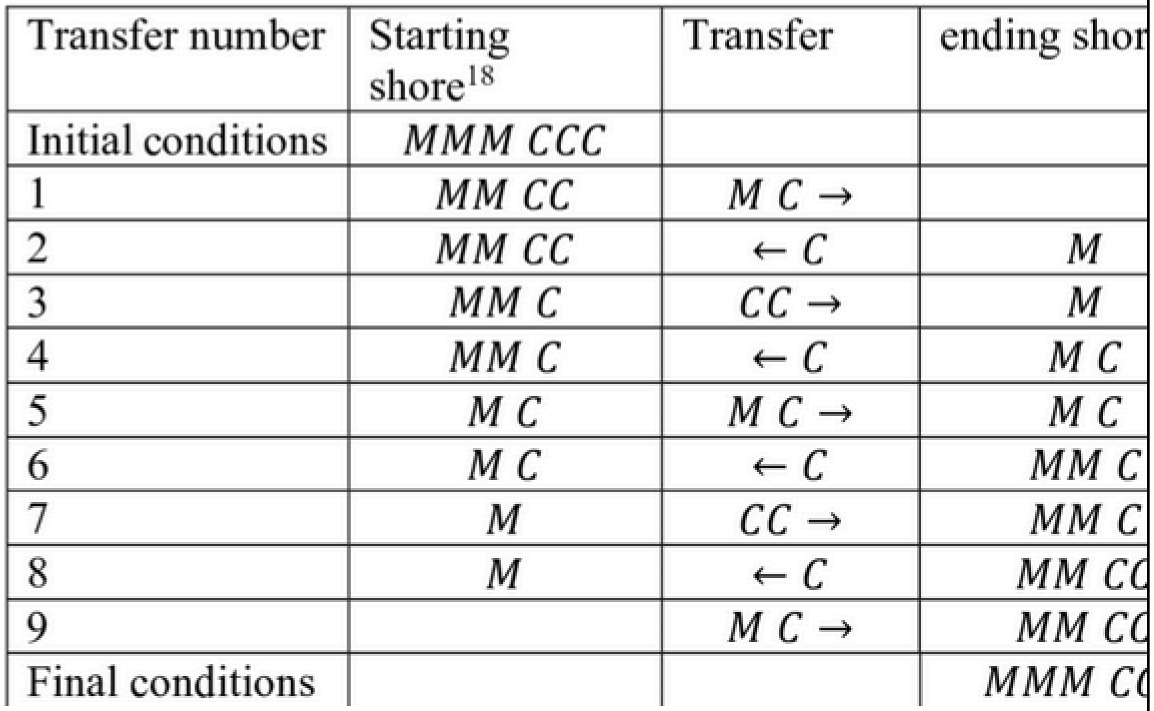
- **Boat Movement**: The boat can be moved from one bank to the other.
- **Passenger Movement**: The boat can carry up to two individuals at a time. The actuators control the loading and unloading of the boat:
    - Selecting which individuals to move (missionaries and/or cannibals) and their destination bank.

Sensors

- **Bank Status**: Sensors monitor the number of missionaries and cannibals on both banks. For this case we'll be using the counter's.
- **Boat Position**: Sensors detect whether the boat is on the left bank or the right bank. Conditional statements are not worth using here.
- **Safety Check**: Sensors ensure that the safety constraint (missionaries not being outnumbered on either bank) is maintained throughout the process.

This PEAS description outlines how the Missionaries and Cannibals problem is structured, specifying what is measured to determine success, the setup of the problem environment, and the mechanisms for action and observation.

Description of the solution through the image.



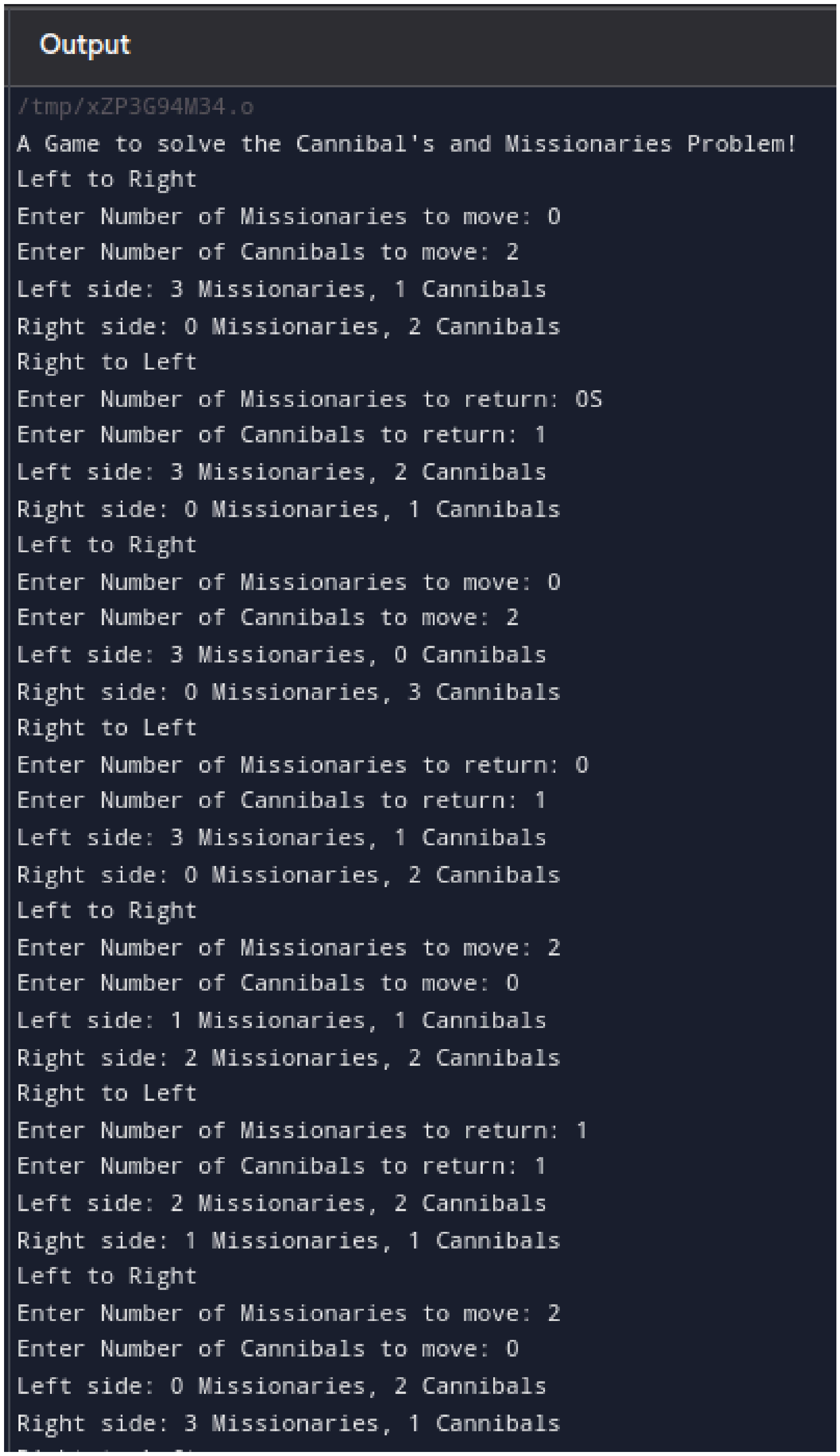
Transfer number	Starting shore[18]	Transfer	ending shor
Initial conditions	$MMM\ CCC$		
1	$MM\ CC$	$M\ C \rightarrow$	
2	$MM\ CC$	$\leftarrow C$	M
3	$MM\ C$	$CC \rightarrow$	M
4	$MM\ C$	$\leftarrow C$	$M\ C$
5	$M\ C$	$M\ C \rightarrow$	$M\ C$
6	$M\ C$	$\leftarrow C$	$MM\ C$
7	M	$CC \rightarrow$	$MM\ C$
8	M	$\leftarrow C$	$MM\ CC$
9		$M\ C \rightarrow$	$MM\ CC$
Final conditions			$MMM\ CC$

Code :

```
#include <stdio.h>

int isValidState(int missionaries, int cannibals) {

    return (missionaries >= 0 || missionaries >= cannibals);

}

int main() {

    printf("A Game to solve the Cannibal's and Missionaries Problem!\n");

    int lCan = 3;
int lMis = 3;

int rCan = 0;

int rMis = 0;

int a, b;

while (rMis != 3 || rCan != 3) {

    if (!isValidState(lMis, lCan) || !isValidState(rMis, rCan)) {

        printf("Error \n");

        return 0;

    }

    printf("Left to Right\n");

    printf("Enter Number of Missionaries to move: ");

    if (scanf("%d", &a) != 1 || a < 0 || a > lMis) {

        printf("Invalid input. Exiting.\n");

        return 1;

    }

    printf("Enter Number of Cannibals to move: ");

    if (scanf("%d", &b) != 1 || b < 0 || b > lCan) {

        printf("Invalid input. Exiting.\n");

        return 1;

    }


    lCan -= b;

    lMis -= a;
```

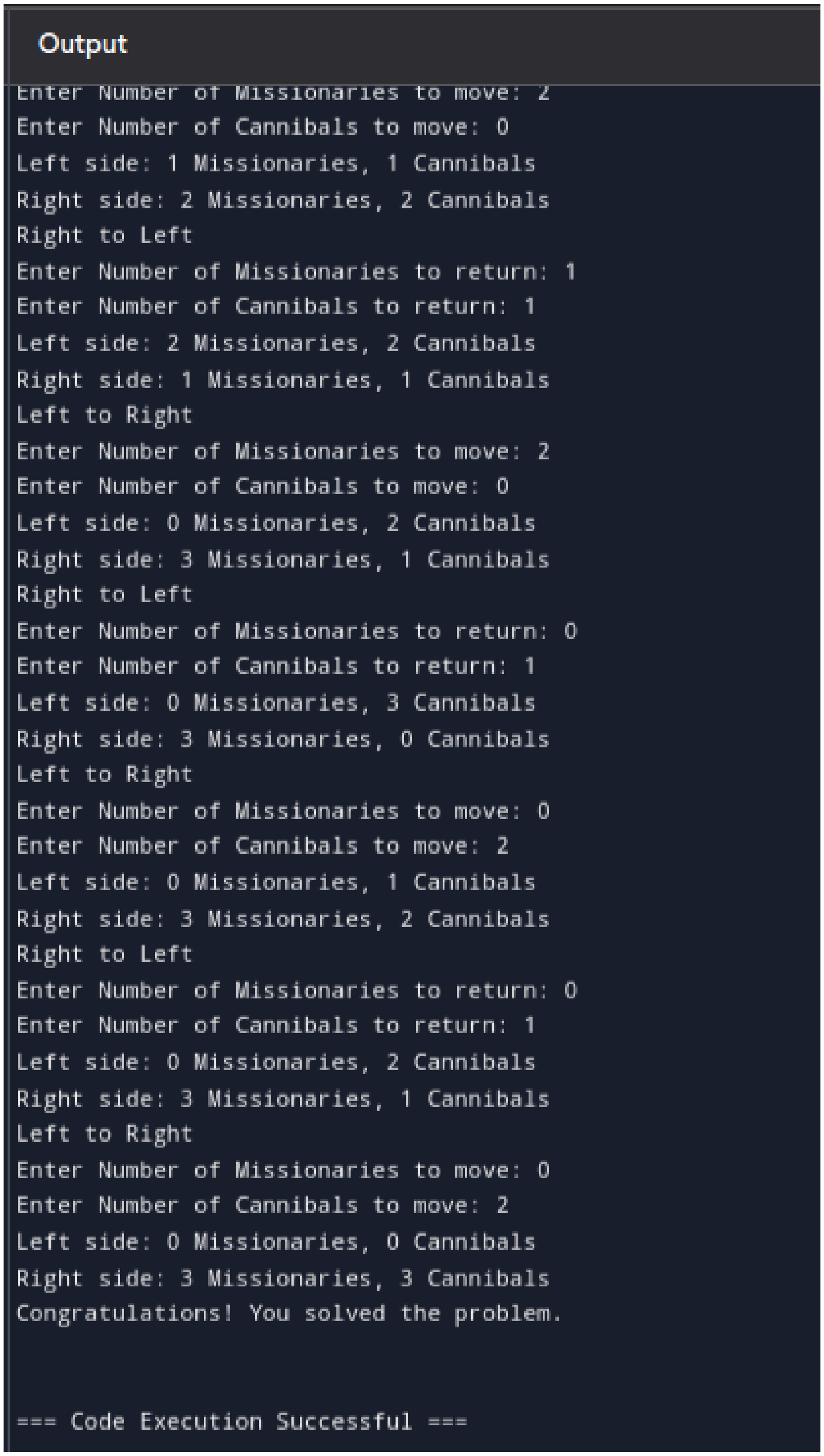
```
rCan += b;
rMis += a;


if (!isValidState(lMis, lCan) || !isValidState(rMis, rCan)) {
    printf("Failure: Game Over! Cannibals ate the missionaries.\n");
    return 0;
}
printf("Left side: %d Missionaries, %d Cannibals\n", lMis, lCan);
printf("Right side: %d Missionaries, %d Cannibals\n", rMis, rCan);
if (rMis == 3 && rCan == 3) {
    printf("Congratulations! You solved the problem.\n");
    return 0;
}
printf("Right to Left\n");
printf("Enter Number of Missionaries to return: ");
if (scanf("%d", &a) != 1 || a < 0 || a > rMis) {
    printf("Invalid input. Exiting.\n");
    return 1;
}
printf("Enter Number of Cannibals to return: ");
if (scanf("%d", &b) != 1 || b < 0 || b > rCan) {
    printf("Invalid input. Exiting.\n");
    return 1;
```

```
        }


        rCan -= b;

        rMis -= a;

        lCan += b;

        lMis += a;


        if (!isValidState(lMis, lCan) || !isValidState(rMis, rCan)) {

            printf("Failure: Game Over! Cannibals ate the missionaries.\n");

            return 0;

        }

        printf("Left side: %d Missionaries, %d Cannibals\n", lMis, lCan);

        printf("Right side: %d Missionaries, %d Cannibals\n", rMis, rCan);

    }

    return 0;

}
```



Output

```
/tmp/xZP3G94M34.o
A Game to solve the Cannibal's and Missionaries Problem!
Left to Right
Enter Number of Missionaries to move: 0
Enter Number of Cannibals to move: 2
Left side: 3 Missionaries, 1 Cannibals
Right side: 0 Missionaries, 2 Cannibals
Right to Left
Enter Number of Missionaries to return: 0S
Enter Number of Cannibals to return: 1
Left side: 3 Missionaries, 2 Cannibals
Right side: 0 Missionaries, 1 Cannibals
Left to Right
Enter Number of Missionaries to move: 0
Enter Number of Cannibals to move: 2
Left side: 3 Missionaries, 0 Cannibals
Right side: 0 Missionaries, 3 Cannibals
Right to Left
Enter Number of Missionaries to return: 0
Enter Number of Cannibals to return: 1
Left side: 3 Missionaries, 1 Cannibals
Right side: 0 Missionaries, 2 Cannibals
Left to Right
Enter Number of Missionaries to move: 2
Enter Number of Cannibals to move: 0
Left side: 1 Missionaries, 1 Cannibals
Right side: 2 Missionaries, 2 Cannibals
Right to Left
Enter Number of Missionaries to return: 1
Enter Number of Cannibals to return: 1
Left side: 2 Missionaries, 2 Cannibals
Right side: 1 Missionaries, 1 Cannibals
Left to Right
Enter Number of Missionaries to move: 2
Enter Number of Cannibals to move: 0
Left side: 0 Missionaries, 2 Cannibals
Right side: 3 Missionaries, 1 Cannibals
```



Output

```
Enter Number of Missionaries to move: 2
Enter Number of Cannibals to move: 0
Left side: 1 Missionaries, 1 Cannibals
Right side: 2 Missionaries, 2 Cannibals
Right to Left
Enter Number of Missionaries to return: 1
Enter Number of Cannibals to return: 1
Left side: 2 Missionaries, 2 Cannibals
Right side: 1 Missionaries, 1 Cannibals
Left to Right
Enter Number of Missionaries to move: 2
Enter Number of Cannibals to move: 0
Left side: 0 Missionaries, 2 Cannibals
Right side: 3 Missionaries, 1 Cannibals
Right to Left
Enter Number of Missionaries to return: 0
Enter Number of Cannibals to return: 1
Left side: 0 Missionaries, 3 Cannibals
Right side: 3 Missionaries, 0 Cannibals
Left to Right
Enter Number of Missionaries to move: 0
Enter Number of Cannibals to move: 2
Left side: 0 Missionaries, 1 Cannibals
Right side: 3 Missionaries, 2 Cannibals
Right to Left
Enter Number of Missionaries to return: 0
Enter Number of Cannibals to return: 1
Left side: 0 Missionaries, 2 Cannibals
Right side: 3 Missionaries, 1 Cannibals
Left to Right
Enter Number of Missionaries to move: 0
Enter Number of Cannibals to move: 2
Left side: 0 Missionaries, 0 Cannibals
Right side: 3 Missionaries, 3 Cannibals
Congratulations! You solved the problem.


=== Code Execution Successful ===
```

Conclusion :	Successfully solving the Missionaries and Cannibals problem demonstrates the importance of managing constraints and systematic planning in problem-solving. It underscores the value of careful strategy development and methodical exploration of possible states to achieve an optimal solution.