VIT Vidyalankar Institute of Technology Accredited A+ by NAAC	**DEPARTMENT OF COMPUTER ENGINEERING**

VIT Vidyalankar Institute of Technology Accredited A+ by NAAC

Experiment No. 6

Semester	T.E. Semester VI
Subject	ARTIFICIAL INTELLIGENCE (CSL 604)
Subject Professor In-charge	Prof. Avinash Shrivas
Assisting Teachers	Prof. Avinash Shrivas

Student Name	Deep Salunkhe
Roll Number	21102A0014
Lab Number	310A

Title:

Implementation of Missionaries and cannibals' problem

Theory:

Problem Statement:
In this problem, there are three missionaries and three cannibals on one side of the river, along with a boat that can carry at most two people. The goal is to transport all missionaries and cannibals to the other side of the river, without ever leaving a group of missionaries in one place outnumbered by the cannibals. If this happens, the cannibals will eat the missionaries, resulting in failure.

Constraints:

1. At most two people (either missionaries or cannibals) can be in the boat for each trip.
2. The number of cannibals must never outnumber the number of missionaries on either side of the river, otherwise, the missionaries will be eaten.

Objective:
The objective is to find a sequence of boat trips that will successfully transport all three missionaries and three cannibals from one side of the river to the other, adhering to the constraints mentioned above.

1. Global Variables: We defined global variables to represent the number of missionaries and cannibals on each side of the river, as well as the boat's position.
2. Functions:
    - print_state(): This function displays the current state of the mission and cannibal positions on both sides of the river, along with the boat's position.
    - is_goal(): This function checks if the goal state has been reached, i.e., all missionaries and cannibals have crossed to the opposite side.
    - handle_a_to_b(): This function handles the movement of missionaries and cannibals from side A to side B.
    - handle_b_to_a(): This function handles the movement of missionaries and cannibals from side B to side A.
3. Main Function: In the main() function, we repeatedly call print_state() to display the current state until the goal state is achieved. We then print a message indicating that the goal has been reached.

Program Code:

```
#include <iostream>
using namespace std;

int side_a[2] = {3, 3};
int side_b[2] = {0, 0};
bool is_boat_on_side_a = true;

void print_state() {
    cout << endl;
    cout << "   Side A            |   Side B" << endl;
    cout << "   Missionaries: " << side_a[0] << "   |   Missionaries: " <<
side_b[0] << endl;
    cout << "   Cannibals: " << side_a[1] << "      |   Cannibals: " << side_b[1]
<< endl;

    if (is_boat_on_side_a) {
        cout << "   Boat here         | " << endl;
    } else {
        cout << "                     |   Boat here" << endl;
    }

    cout << endl;
}

bool is_goal() {
    return side_b[0] == 3 && side_b[1] == 3 && !is_boat_on_side_a;
}

void handle_a_to_b() {
```

```
    int missionaries, cannibals;

    while (true) {
        cout << "Enter number of missionaries travelling to side b: ";
        cin >> missionaries;
        cout << "Enter number of cannibals travelling to side b: ";
        cin >> cannibals;

        if (!(cannibals >= -1 && cannibals <= 2) || !(missionaries >= -1 && mis-
sionaries <= 2) || missionaries + cannibals > 2
            || missionaries > side_a[0] || cannibals > side_a[1] || (side_a[0] -
missionaries != 0 && side_a[1] - cannibals > side_a[0] - missionaries)
            || (side_b[0] + missionaries != 0 && side_b[1] + cannibals >
side_b[0] + missionaries)) {
            cout << "Invalid input!" << endl;
        } else {
            break;
        }
    }

    side_a[0] -= missionaries;
    side_a[1] -= cannibals;
    side_b[0] += missionaries;
    side_b[1] += cannibals;
    is_boat_on_side_a = false;
}

void handle_b_to_a() {
    int missionaries, cannibals;

    while (true) {
        cout << "Enter number of missionaries travelling to side a: ";
        cin >> missionaries;
        cout << "Enter number of cannibals travelling to side a: ";
        cin >> cannibals;

        if (!(cannibals >= -1 && cannibals <= 2) || !(missionaries >= -1 && mis-
sionaries <= 2) || missionaries + cannibals > 2
            || missionaries > side_b[0] || cannibals > side_b[1] || (side_b[0] -
missionaries != 0 && side_b[1] - cannibals > side_b[0] - missionaries)
            || (side_a[0] + missionaries != 0 && side_a[1] + cannibals >
side_a[0] + missionaries)) {
            cout << "Invalid input!" << endl;
        } else {
            break;
        }
    }
```

```
    side_b[0] -= missionaries;
    side_b[1] -= cannibals;
    side_a[0] += missionaries;
    side_a[1] += cannibals;
    is_boat_on_side_a = true;
}

int main() {
    while (!is_goal()) {
        print_state();

        if (is_boat_on_side_a) {
            handle_a_to_b();
        } else {
            handle_b_to_a();
        }
    }

    print_state();
    cout << "Goal reached!" << endl;

    return 0;
}
```

Output:

```
PS E:\GIt\SEM-6> cd "e:\GIt\SEM-6\AI\" ; if ($?) { g++ Lab6_A.cpp -o Lab6_A } ; if ($?) { .\Lab6_A }

   Side A            |   Side B
   Missionaries: 3   |   Missionaries: 0
   Cannibals: 3      |   Cannibals: 0
   Boat here         |

Enter number of missionaries travelling to side b: 1
Enter number of cannibals travelling to side b: 1

   Side A            |   Side B
   Missionaries: 2   |   Missionaries: 1
   Cannibals: 2      |   Cannibals: 1
                     |   Boat here

Enter number of missionaries travelling to side a: 1
Enter number of cannibals travelling to side a: 0

   Side A            |   Side B
   Missionaries: 3   |   Missionaries: 0
   Cannibals: 2      |   Cannibals: 1
   Boat here         |

Enter number of missionaries travelling to side b: 0
Enter number of cannibals travelling to side b: 2
```

```
   Side A            |   Side B
   Missionaries: 3   |   Missionaries: 0
   Cannibals: 0      |   Cannibals: 3
                     |   Boat here

Enter number of missionaries travelling to side a: 0
Enter number of cannibals travelling to side a: 1

   Side A            |   Side B
   Missionaries: 3   |   Missionaries: 0
   Cannibals: 1      |   Cannibals: 2
   Boat here         |

Enter number of missionaries travelling to side b: 2
Enter number of cannibals travelling to side b: 0

   Side A            |   Side B
   Missionaries: 1   |   Missionaries: 2
   Cannibals: 1      |   Cannibals: 2
                     |   Boat here

Enter number of missionaries travelling to side a: 1
Enter number of cannibals travelling to side a: 1

   Side A            |   Side B
   Missionaries: 2   |   Missionaries: 1
   Cannibals: 2      |   Cannibals: 1
```

```
Enter number of missionaries travelling to side b: 2
Enter number of cannibals travelling to side b: 0

   Side A             |   Side B
   Missionaries: 0    |   Missionaries: 3
   Cannibals: 2       |   Cannibals: 1
                      |   Boat here

Enter number of missionaries travelling to side a: 0
Enter number of cannibals travelling to side a: 1

   Side A             |   Side B
   Missionaries: 0    |   Missionaries: 3
   Cannibals: 3       |   Cannibals: 0
   Boat here          |

Enter number of missionaries travelling to side b: 0
Enter number of cannibals travelling to side b: 2

   Side A             |   Side B
   Missionaries: 0    |   Missionaries: 3
   Cannibals: 1       |   Cannibals: 2
                      |   Boat here

Enter number of missionaries travelling to side a: 1
Enter number of cannibals travelling to side a: 0
```

```
Enter number of missionaries travelling to side a: 1
Enter number of cannibals travelling to side a: 0

   Side A            |   Side B
   Missionaries: 1   |   Missionaries: 2
   Cannibals: 1      |   Cannibals: 2
   Boat here         |

Enter number of missionaries travelling to side b: 1
Enter number of cannibals travelling to side b: 1

   Side A            |   Side B
   Missionaries: 0   |   Missionaries: 3
   Cannibals: 0      |   Cannibals: 3
                     |   Boat here

Goal reached!
PS E:\GIt\SEM-6\AI>
```

Conclusion:

In conclusion, we successfully implemented a solution for the missionary-cannibal problem using C++. The problem required careful consideration of constraints to ensure the safety of the missionaries and cannibals during transportation across the river. Our solution effectively handled the movement of missionaries and cannibals between the two sides of the river while adhering to the specified constraints. Through this implementation, we gained practical experience in problem-solving and algorithmic thinking, which are essential skills in the field of artificial intelligence and computer science.