Constraint Satisfaction Problem

Name: Bhavin Patil
Roll No.: 66

Code:

```cpp
#include <bits/stdc++.h>
using namespace std;
int const N = 9;
int const n = 3;

// printing board
void printBoard(vector<vector<int>> board)
{
   cout << "+++++++++++++++++++++" << endl;
   for (int i = 0; i < N; i++)
   {
       for (int j = 0; j < N; j++)
       {
           cout << board[i][j] << " ";
       }
       cout << endl;
   }
   cout << "+++++++++++++++++++++" << endl;
}

// check row
bool checkRow(vector<vector<int>> board)
{
   for (int i = 0; i < N; i++)
   {
       int arr[N];
       for (int j = 0; j < N; j++)
           arr[j] = board[i][j];

       sort(arr, arr + N);
       for (int j = 1; j < N; j++)
       {
           if (arr[j] == 0)
```

```
                continue;
            if (arr[j] == arr[j - 1])
                return false;
        }
    }
    return true;
}

// check col
bool checkCol(vector<vector<int>> board)
{
    for (int i = 0; i < N; i++)
    {
        int arr[N];
        for (int j = 0; j < N; j++)
            arr[j] = board[j][i];

        sort(arr, arr + N);
        for (int j = 1; j < N; j++)
        {
            if (arr[j] == 0)
                continue;
            if (arr[j] == arr[j - 1])
                return false;
        }
    }
    return true;
}

// check box
bool checkBox(vector<vector<int>> board)
{
    for (int i = 0; i < N; i++)
    {
        // create array
        int arr[N];
        for (int j = 0; j < N; j++)
            arr[j] = board[((i / n) * n) + (j / n)][((i % n) * n) + (j %
n)];
        // evaluating array
```

```
        sort(arr, arr + N);
        for (int j = 1; j < N; j++)
        {
            if (arr[j] == 0)
                continue;
            if (arr[j] == arr[j - 1])
                return false;
        }
    }
    return true;
}

// check Board condition
bool checkBoard(vector<vector<int>> board)
{
    if (checkRow(board) && checkCol(board) && checkBox(board))
        return true;
    else
        return false;
}

int findNextSpace(vector<vector<int>> board)
{
    for (int i = 0; i < N * N; i++)
    {
        if (board[i / N][i % N] == 0)
            return i;
    }
    return -1;
}

void insertBoard(vector<vector<int>> &board, int index, int num)
{
    board[index / N].at(index % N) = num;
}

bool canInsertBoard(vector<vector<int>> board, int nextAvailabelSpace, int
num)
{
```

```
    insertBoard(board, nextAvailabelSpace, num);
    if (checkBoard(board))
        return true;
    else
        return false;
}

bool solveBoard(vector<vector<int>> &board)
{
    // sleep(1);
    // printBoard(board);
    int nextAvailabaleSpace = findNextSpace(board);
    if (nextAvailabaleSpace == -1)
        return true;
    for (int i = 1; i <= N; i++)
    {
        if (canInsertBoard(board, nextAvailabaleSpace, i))
        {
            insertBoard(board, nextAvailabaleSpace, i);
            if (solveBoard(board))
                return true;
            else
                insertBoard(board, nextAvailabaleSpace, 0);
        }
    }
    return false;
}

int main()
{
    vector<vector<int>> board(N);

    // filling the board
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            int input;
            cin >> input;
            if (input >= 0 && input <= N)
```

```
            {
                board[i].push_back(input);
            }
            else
            {
                cout << "Invalid Input!";
                return -1;
            }
        }
    }
    cout << endl;
    if (solveBoard(board))
    {
        printBoard(board);
    }
    else
        cout << "This is board in Invalid or Impossible to solve!" << endl;
}
```

Output:

```
cd "/home/bhavin/Documents/GitHub/Artigficial-Intelligence-Problems/Contraint Satisfaction Problem/" && g++ sudokuSolver.cpp -o sudokuSolver && "/home/bhavin/Documents/GitHub/Artigficial-Intelligence-Problems/Contraint Satisfaction Problem/"sudokuSolver
bhavin@bhavin-Predator-PH315-53:~/Documents/GitHub/Artigficial-Intelligence-Problems/Contraint Satisfaction Problem$ cd "/home/bhavin/Documents/GitHub/Artigficial-Intelligence-Problems/Contraint Satisfaction Problem/" && g++ sudokuSolver.cpp -o sudokuSolver && "/home/bhavin/Documents/GitHub/Artigficial-Intelligence-Problems/Contraint Satisfaction Problem/"sudokuSolver
1 0 0 5 0 0 0 0 3
0 3 7 0 0 1 0 0 4
0 0 0 0 0 2 0 0 0
0 0 0 0 9 4 0 0 0
0 0 4 0 0 0 0 3 2
0 0 0 3 2 5 0 0 0
0 0 0 0 0 0 2 0 0
7 4 0 0 0 0 0 0 0
8 0 0 0 0 3 1 0 0

++++++++++++++++++++
1 2 6 5 4 7 8 9 3
9 3 7 6 8 1 5 2 4
4 5 8 9 3 2 6 1 7
2 1 3 8 9 4 7 5 6
5 8 4 1 7 6 9 3 2
6 7 9 3 2 5 4 8 1
3 6 1 4 5 8 2 7 9
7 4 5 2 1 9 3 6 8
8 9 2 7 6 3 1 4 5
++++++++++++++++++++
bhavin@bhavin-Predator-PH315-53:~/Documents/GitHub/Artigficial-Intelligence-Problems/Contraint Satisfaction Problem$
```