

```

#include <iostream>
using namespace std;

const int N = 9;
void printGrid(int grid[N][N]) {
    for (int row = 0; row < N; row++) {
        for (int col = 0; col < N; col++) {
            cout << grid[row][col] << " ";
        }
        cout << endl;
    }
}

bool isSafe(int grid[N][N], int row, int col, int num) {

    for (int x = 0; x < N; x++) {
        if (grid[row][x] == num || grid[x][col] == num)
            return false;
    }

    int startRow = row - row % 3, startCol = col - col % 3;
    for (int r = 0; r < 3; r++)
        for (int c = 0; c < 3; c++)
            if (grid[startRow + r][startCol + c] == num)
                return false;

    return true;
}

bool solveSudoku(int grid[N][N]) {
    for (int row = 0; row < N; row++) {
        for (int col = 0; col < N; col++) {

            if (grid[row][col] == 0) {

                for (int num = 1; num <= 9; num++) {
                    if (isSafe(grid, row, col, num)) {
                        grid[row][col] = num;
                        if (solveSudoku(grid))
                            return true;
                        grid[row][col] = 0;
                    }
                }
            }
        }
    }
}

```

```

        return false;
    }
}
return true;
}

int main() {
    int grid[N][N] = {
        {5, 3, 0, 0, 7, 0, 0, 0, 0},
        {6, 0, 0, 1, 9, 5, 0, 0, 0},
        {0, 9, 8, 0, 0, 0, 0, 6, 0},
        {8, 0, 0, 0, 6, 0, 0, 0, 3},
        {4, 0, 0, 8, 0, 3, 0, 0, 1},
        {7, 0, 0, 0, 2, 0, 0, 0, 6},
        {0, 6, 0, 0, 0, 0, 2, 8, 0},
        {0, 0, 0, 4, 1, 9, 0, 0, 5},
        {0, 0, 0, 0, 8, 0, 0, 7, 9}
    };

    cout << "Initial Sudoku Grid:\n";
    printGrid(grid);

    if (solveSudoku(grid)) {
        cout << "\nSolved Sudoku Grid:\n";
        printGrid(grid);
    } else {
        cout << "\nNo solution exists for the given Sudoku.\n";
    }

    return 0;
}

```